طراحی الگوریتم ها

فصل ۵: ساختارهای پیشرفته

لیلا ناموری تازه‌کند

نیمسال دوم سال تحصیلی ۱۴۰۴-۱۴۰۳

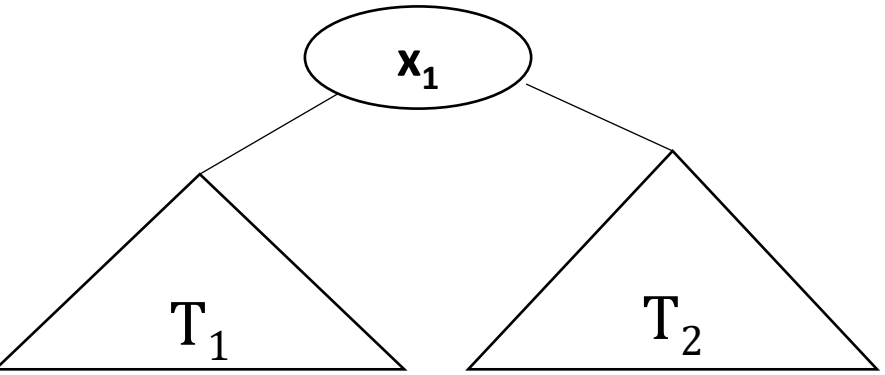
مطالب فصل

- درخت B (B-tree)
- مجموعه های مجزا (Disjoint Sets)
- یادآوری مبحث هرم (Heap)
- درخت های دوجمله ای (Binomial Trees)
- هرم های دوجمله ای (Binomial Heaps)
- هرم های فیبوناچی (Fibonacci Heaps)

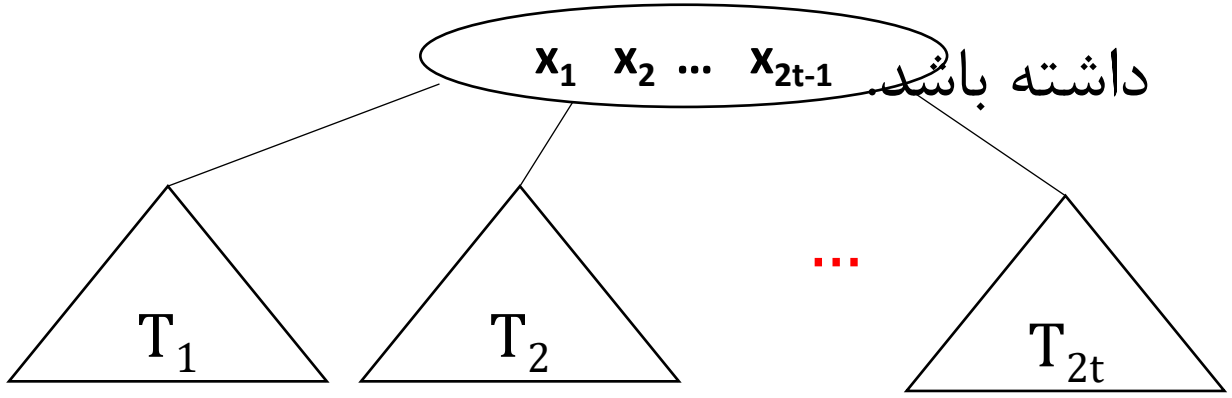
درخت B (B-tree)

- ✓ تعریف: درخت B داده ساختاری است که داده ها را به صورت مرتب شده نگهداری می کند و عملیات جستجو، درج و حذف از مرتبه O(logn) می باشد.
- ✓ درخت B معمولا برای سیستم هایی بکار می رود که دارای بلاک های عظیم اطلاعات هستند.
- ✓ یک درخت B از مرتبه t دارای خواص زیر است (t>=2):

۱- ریشه حداقل دو فرزند و حداکثر 2t فرزند دارد بنابراین باید حداقل ۱ کلید و حداکثر 2t-1 کلید داشته باشد.



$T_1 < x_1 < T_2$



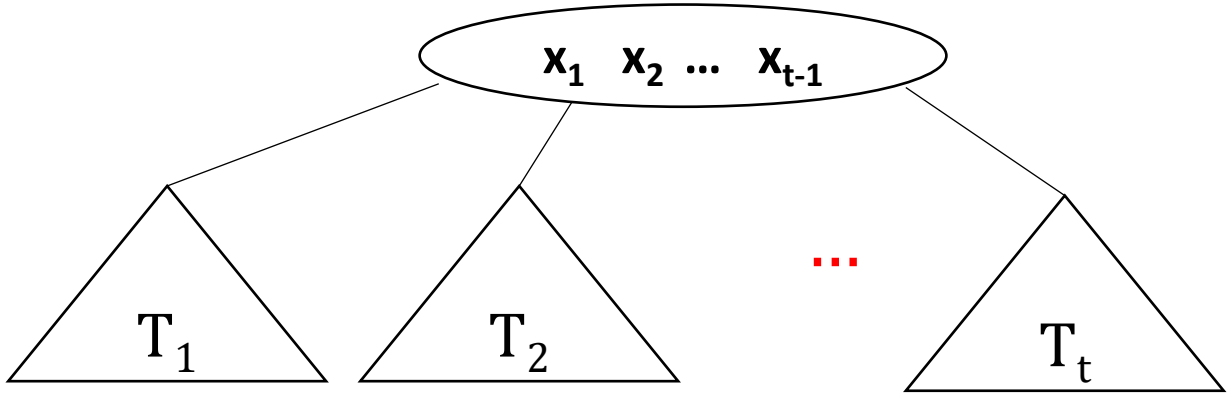
$T_1 < x_1 < T_2 < x_2 < \ldots < x_{2t-1} < T_{2t}$

۲- هر گره به غیر از ریشه دارای حداقل t فرزند و حداکثر 2t فرزند است بنابراین باید حداقل t-1 کلید و حداکثر 2t-1 کلید داشته باشد.

۳- تمام برگ ها در سطح آخر قرار دارند.

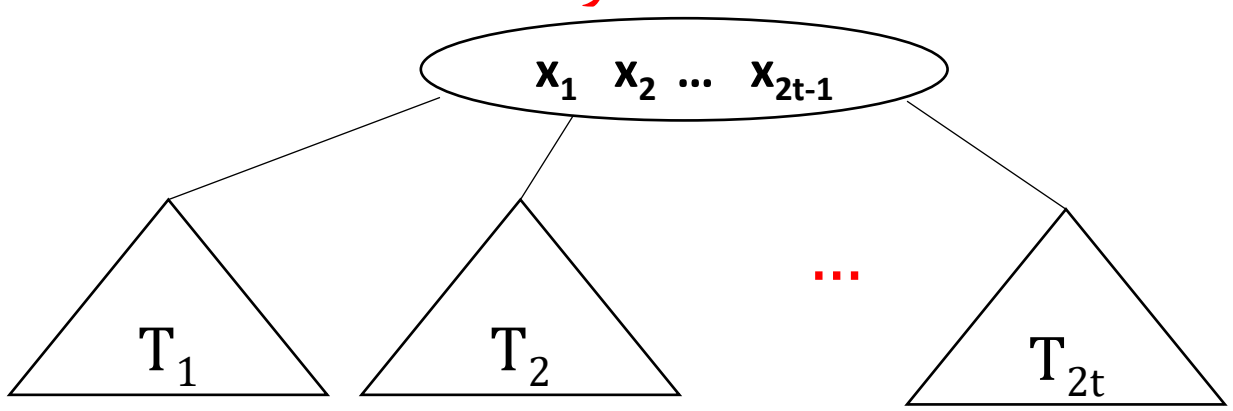
۴- برای هر گره از مرتبه t داریم:

حداقل



$$T_1 < x_1 < T_2 < x_2 < \ldots < x_{t-1} < T_t$$

حداکثر



$$T_1 < x_1 < T_2 < x_2 < \ldots < x_{2t-1} < T_{2t}$$

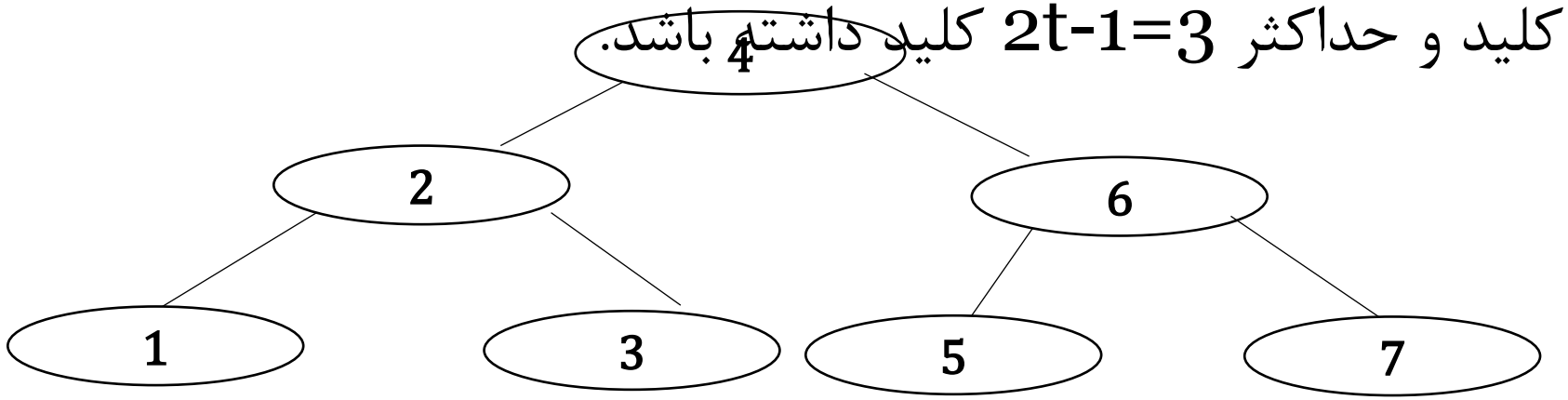
✓ مثال: درخت B از مرتبه $t=2$ و با ارتفاع 2 با حداقل فرزندان و حداکثر فرزندان رسم کنید ضمناً آن را با اعداد دلخواه پر کنید.

جواب:

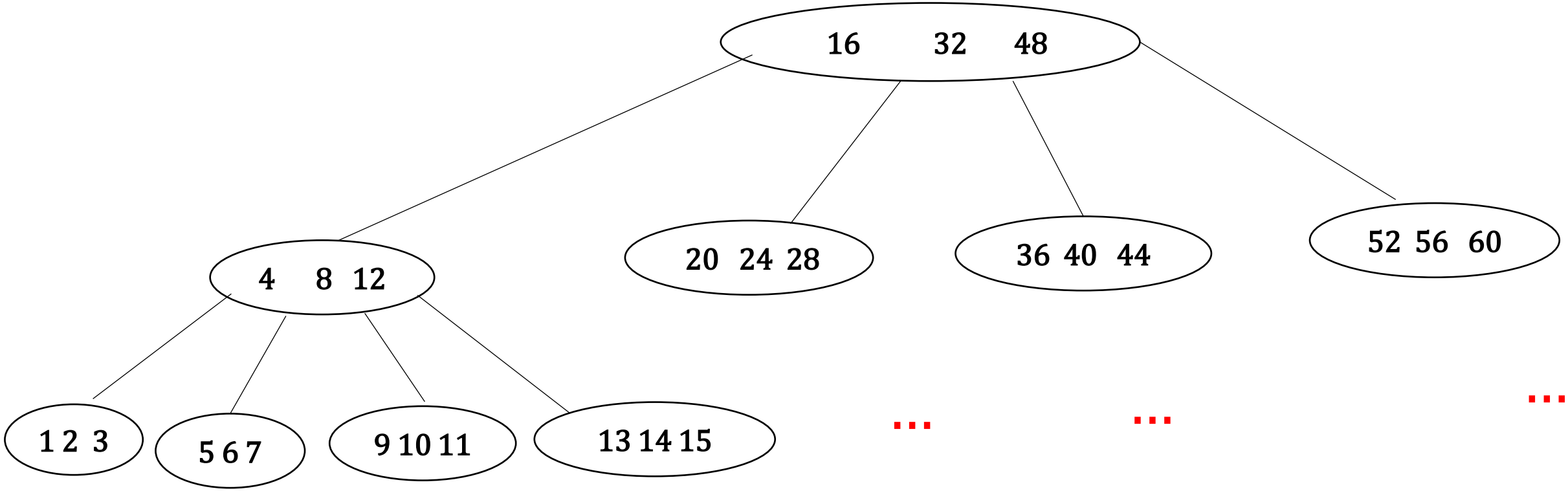
۱- ریشه حداقل دو فرزند دارد و حداکثر $2t=4$ فرزند دارد بنابراین باید حداقل ۱ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

۲- هر گره به غیر از ریشه دارای حداقل $t=2$ فرزند و حداکثر $2t=4$ فرزند است بنابراین باید حداقل $t-1=1$ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

درخت حداقل:

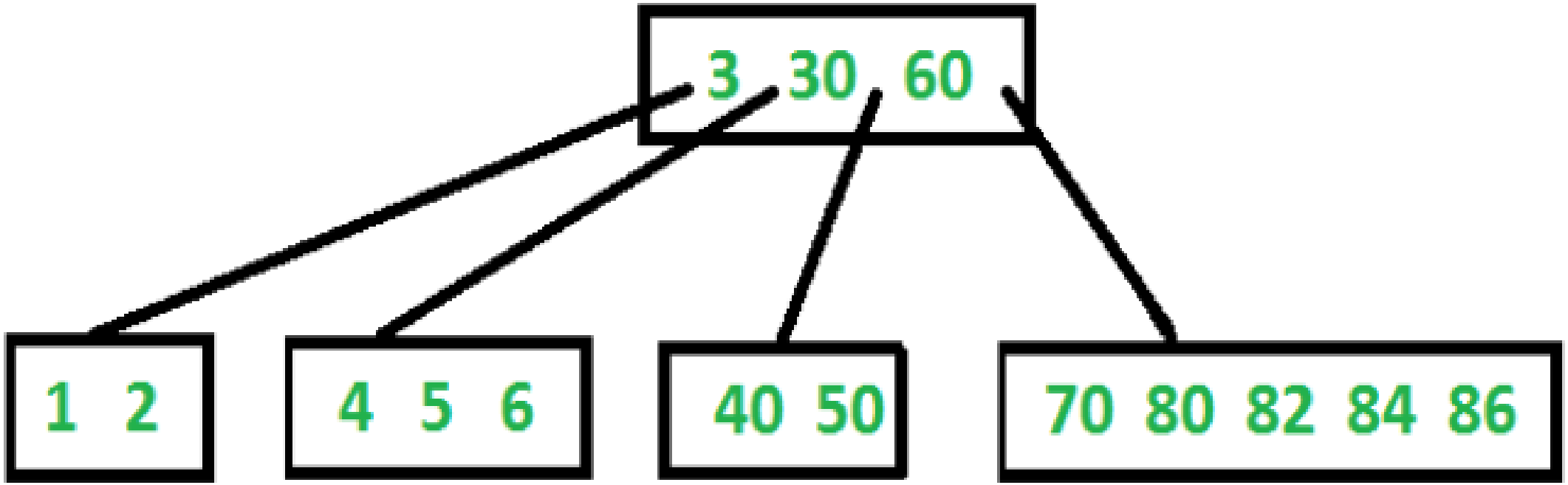


درخت حداکثر:



در واقع، از اعداد ۱ تا ۶۳ را در این درخت نگهداری کرده ایم.

✓ کار در کلاس: آیا درخت زیر می تواند یک درخت B باشد و مقدار t را پیدا کنید؟



✓ جواب : درخت B از مرتبه $t=3$ می باشد چون حداقل کلید هر گره برابر $t-1=2$ و حداکثر کلید هر گره برابر $2t-1=5$ می باشد.

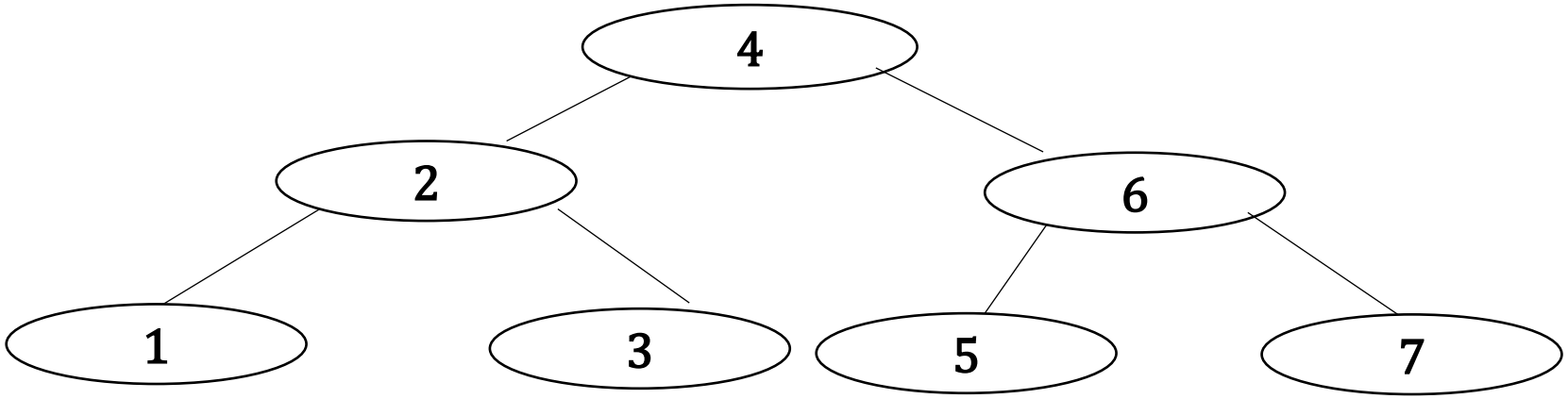
✓ کار در کلاس: حداقل و حداکثر تعداد گره های درخت B با درجه $t=2$ و ارتفاع $h=2$ را محاسبه کنید.

۱- ریشه حداقل دو فرزند دارد و حداکثر $2t=4$ فرزند دارد بنابراین باید حداقل ۱ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

۲- هر گره به غیر از ریشه دارای حداقل $t=2$ فرزند و حداکثر $2t=4$ فرزند است بنابراین باید حداقل $t-1=1$ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

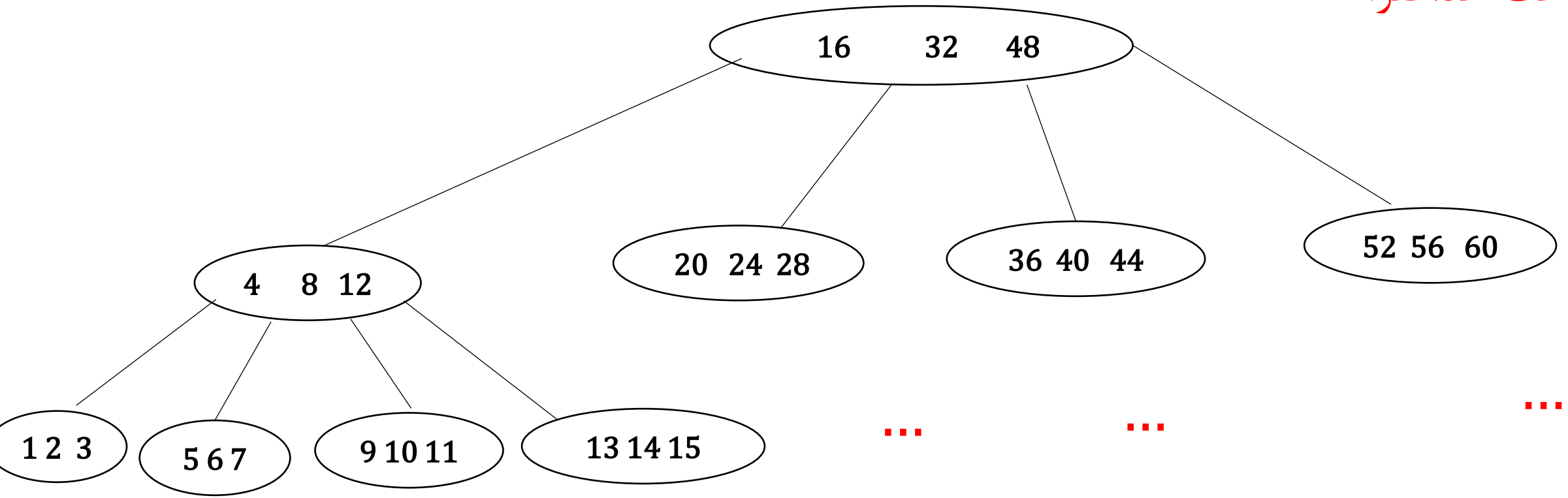
درخت حداقل:

.



$N=1+2+2t=1+2+2*2=7$

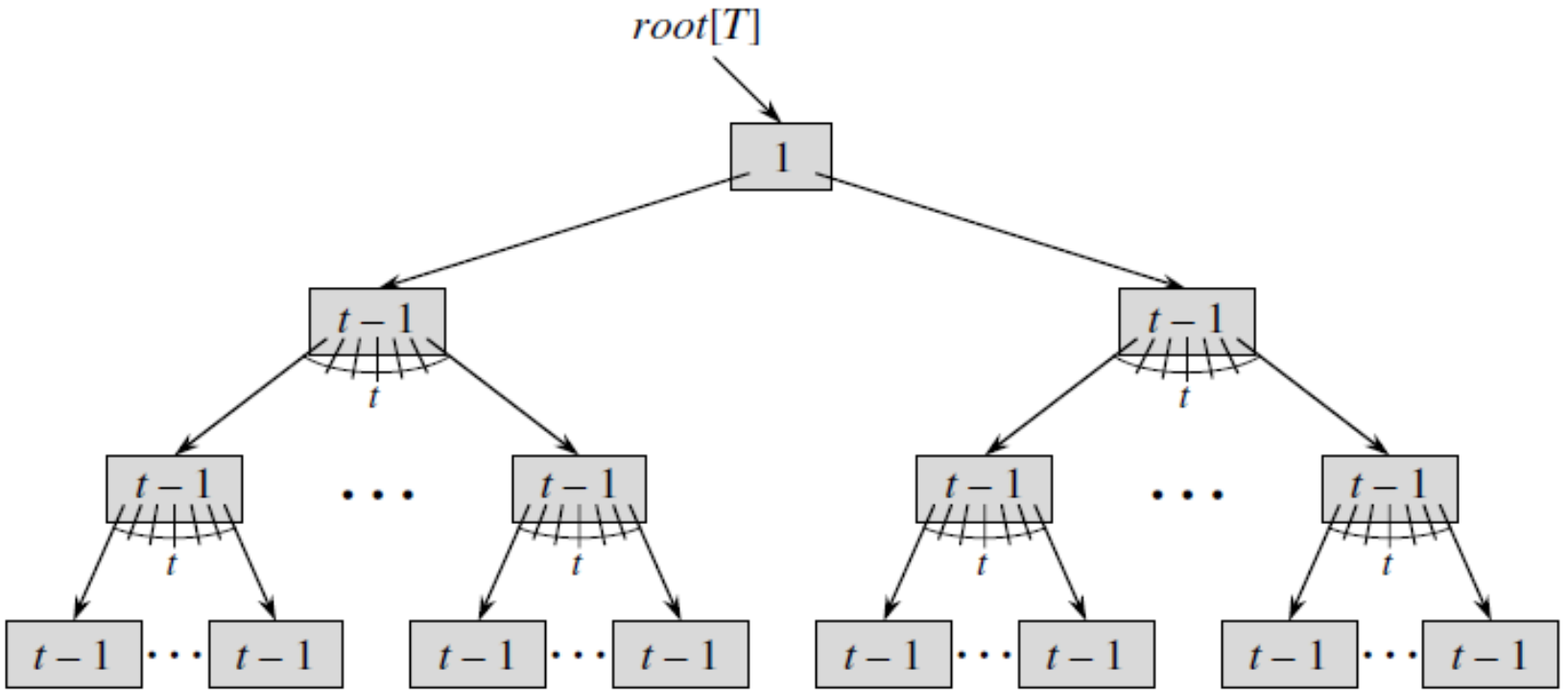
درخت حداکثر:



$$N=1+1*2t+2t*2t=1+2*2+2*2*2*2=1+4+16=21$$

✓ مثال: حداقل و حداکثر تعداد گره های درخت B با درجه t و ارتفاع h را محاسبه کنید.

حداقل: موقعی است که ریشه فقط دو فرزند داشته باشد و بقیه گره های داخلی دقیقا t فرزند داشته باشند.
شکل زیر درخت B را برای h=3 نشان می دهد.



سطح یا عمق	تعداد گره ها
0	1
1	2
2	$2t$
3	$2t^2$

$$N=1+2+2t+2t^2+\ldots+2t^{h-1}=1+2(1+t+t^{2\ldots+}t^{h-1})=1+2*\frac{t^h-1}{t-1}$$

حداکثر: موقعی است که ریشه و بقیه گره های داخلی دقیقا $2t$ فرزند داشته باشند.

در سطح (عمق) صفر فقط یک گره وجود دارد. در سطح ۱ تعداد $2t$ گره وجود دارد. در سطح ۲ تعداد $(2t)^2$ گره وجود دارد. .... در سطح h تعداد $(2t)^h$ گره وجود دارد.

$$N=(2t)^0+(2t)^1+(2t)^2+\dots+(2t)^h=\frac{(2t)^{h+1}-1}{2t-1}$$

کار در کلاس: اگر $t=2$ و $h=2$ باشند حداقل و حداکثر گره ها را محاسبه کنید؟

$$1+2*\frac{t^h-1}{t-1} \le N \le \frac{(2t)^{h+1}-1}{2t-1}$$

$$1+2*\frac{2^2-1}{2-1} \le N \le \frac{(2*2)^{2+1}-1}{2*2-1}$$

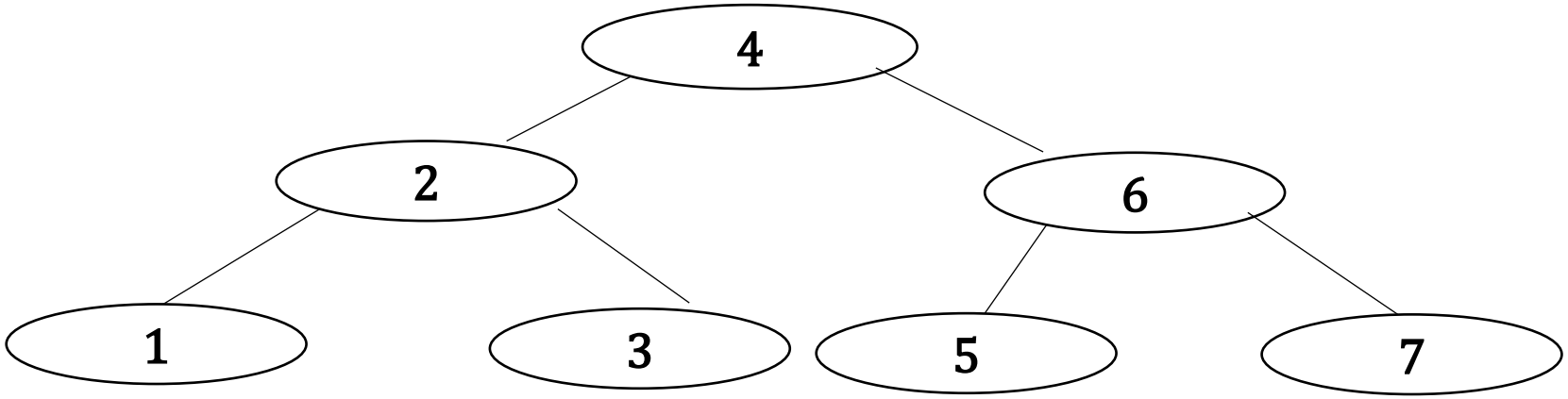
$$7 \le N \le 21$$

✓ مثال: حداقل و حداکثر تعداد کلید های درخت B با درجه $t=2$ و ارتفاع $h=2$ را محاسبه کنید.

۱- ریشه حداقل دو فرزند دارد و حداکثر $2t=4$ فرزند دارد بنابراین باید حداقل ۱ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

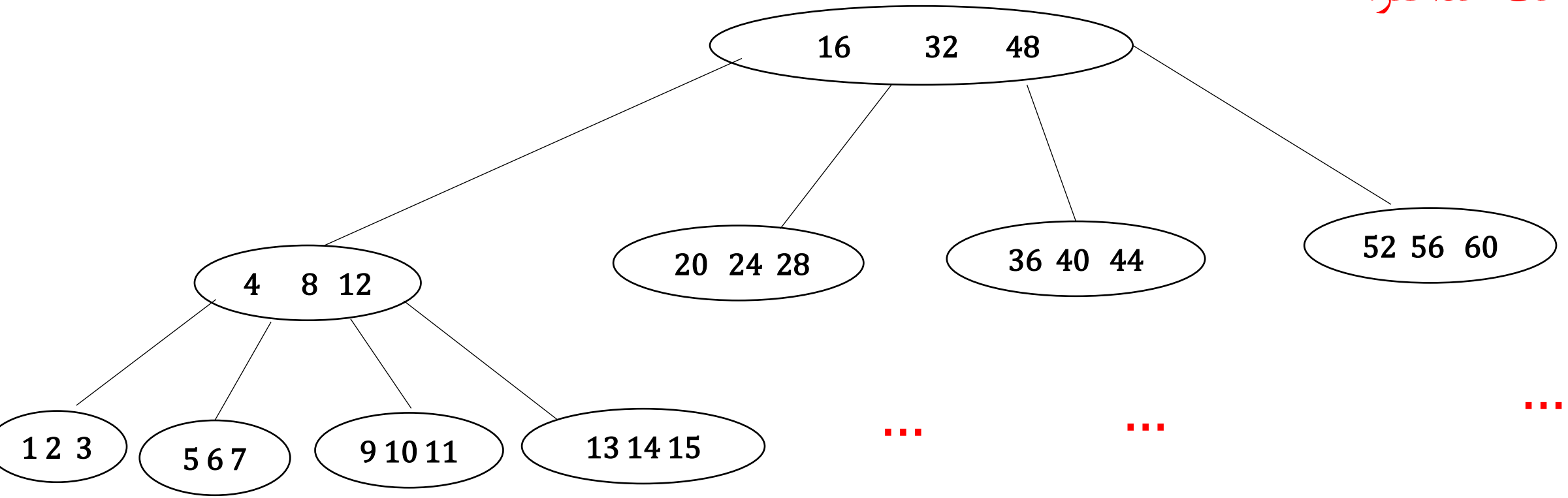
۲- هر گره به غیر از ریشه دارای حداقل $t=2$ فرزند و حداکثر $2t=4$ فرزند است بنابراین باید حداقل $t-1=1$ کلید و حداکثر $2t-1=3$ کلید داشته باشد.

درخت حداقل:



$KN=1*1+2*1+2*t*1=1+2+2*2=7$

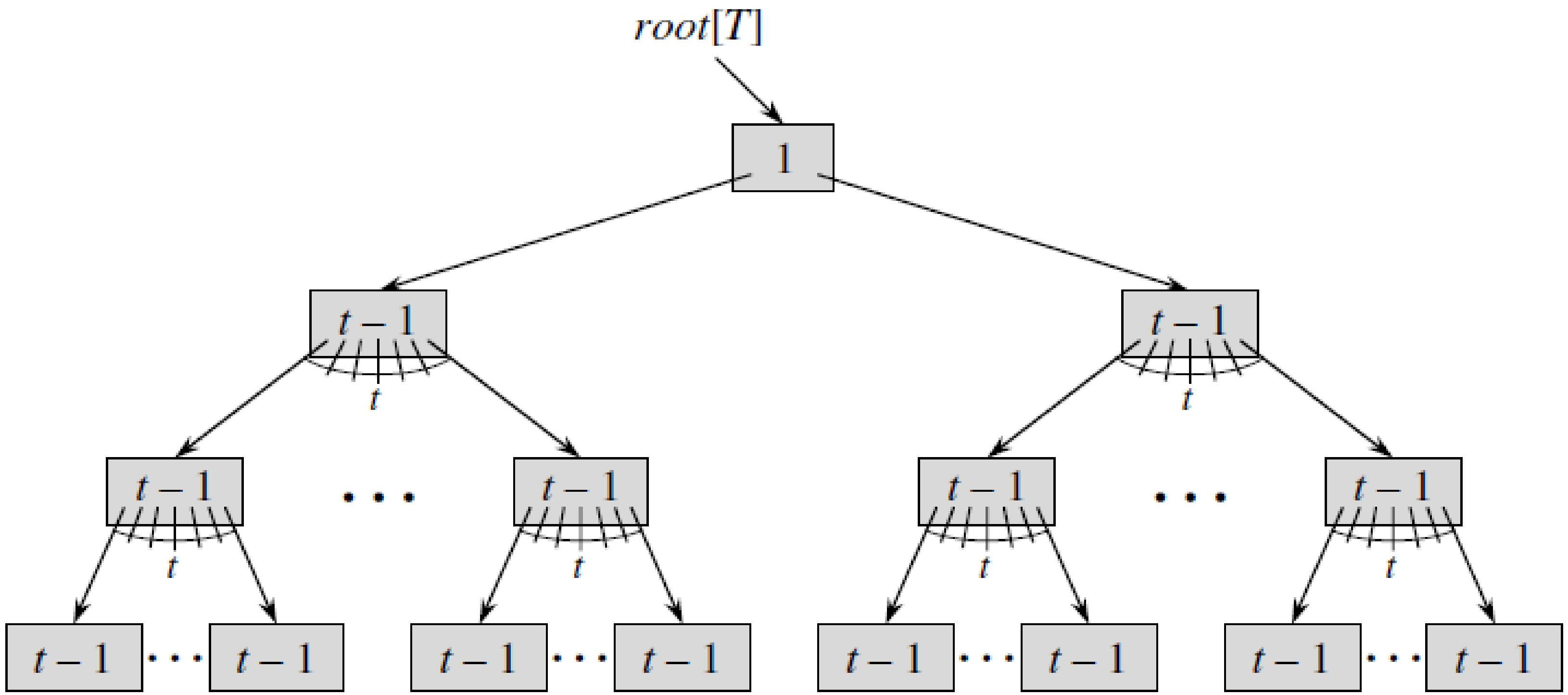
درخت حداکثر:



$$KN=1*3+2t*3+2t*2t*3=(1+2*2+2*2*2*2)*3=(1+4+16)*3=63$$

✓ مثال: حداقل و حداکثر کلید های درخت B با درجه t و ارتفاع h را محاسبه کنید.

حداقل: موقعی است که ریشه فقط یک کلید داشته باشد و بقیه گره های داخلی دقیقا t-1 کلید داشته باشند. شکل زیر درخت B را برای h=3 نشان می دهد.



تعداد کلیدها	سطح یا عمق
$1*1$	0
$2*(t-1)$	1
$2t*(t-1)$	2
$2t^{2}*(t-1)$	3

$$KN=1*1+2*(t-1)+2t*(t-1)+2t^{2}*(t-1)+\ldots+2t^{h-1}*(t-1)$$

$$=1+2(t-1)(1+t+t^{2}\ldots+t^{h-1})=1+2(t-1)*\frac{t^{h}-1}{t-1}=2t^{h}-1$$

حداکثر: موقعی است که ریشه و بقیه گره های داخلی دقیقا 2t-1 کلید داشته باشند.

کافی است تعداد کل گره ها را ضربدر 2t-1 کنیم. بنابراین داریم:

$$KN = N*(2t-1) = \frac{(2t)^{h+1}-1}{2t-1}*(2t-1) = (2t)^{h+1}-1$$

کار در کلاس: اگر $t=2$ و $h=2$ باشند حداقل و حداکثر کلید ها چیست؟

$$2t^{h}-1 \leq KN \leq (2t)^{h+1}-1$$

$$2*2^{2}-1 \leq KN \leq (2*2)^{2+1}-1$$

$$7 \leq KN \leq 63$$

نتیجه: اگر t درجه درخت B و n تعداد کل کلیدهای آن باشد داریم:

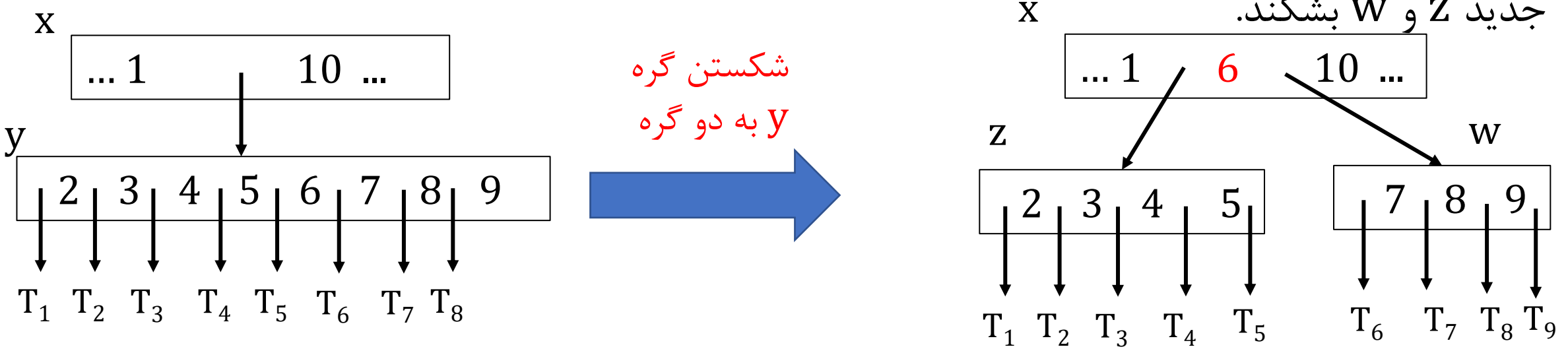
$$h \leq \log_t \frac{n+1}{2}$$

مثال: می توان حدود $n=1000000000$ کلید را در یک درخت B با درجه $t=1000$ و ارتفاع $h=3$ نگهداری کرد.

درج در درخت B (B-tree)

✓ برای درج x در یک درخت B با درجه t، ابتدا جای آن را پیدا کرده و درج می کنیم (همیشه به برگ ها اضافه می شود). اگر تعداد کلیدهای گره حاصله بیشتر از 2t-1 باشد (سرریز: overflow) باید آن را به دو گره بشکنیم.

✓ بعنوان مثال: در قسمتی از درخت B با درجه t=4 ، فرضاً عنصر جدید یعنی ۵ به گره y اضافه شده باشد چون تعداد کلیدهای هر گره حداکثر می تواند 2t-1=7 باشد بنابراین، گره y باید به دو گره جدید z و w بشکند.



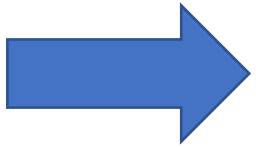
✓ چون تعداد فرزندان گره x یک واحد افزایش پیدا کرد بنابراین به تعداد کلیدهای گره x باید یک واحد اضافه بشود و میانه کلیدهای گره y یعنی 6 به گره x منتقل می شود.

مثال: درخت B با درجه t=2 با عناصر زیر ایجاد کنید (مرحله به مرحله)

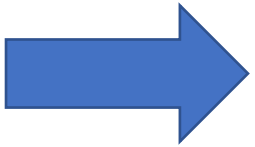
5, 3, 21, 9, 1, 13, 2, 7, 10, 12, 4, 8

جواب: هر گره حداکثر می تواند 2t-1=3 کلید و 2t=4 فرزند داشته باشد.

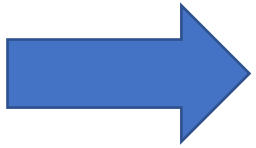
درج عدد ۵



درج عدد ۳

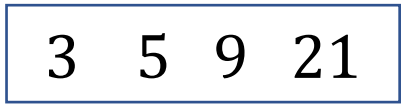


درج عدد ۲۱



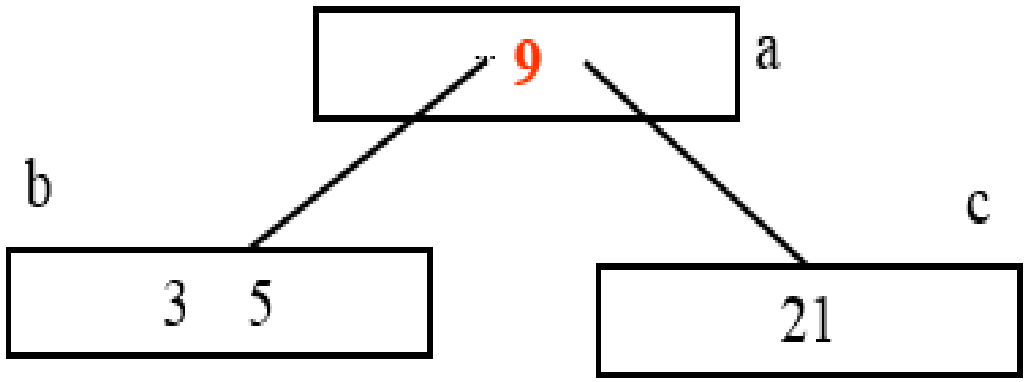
درج عدد ۹

a

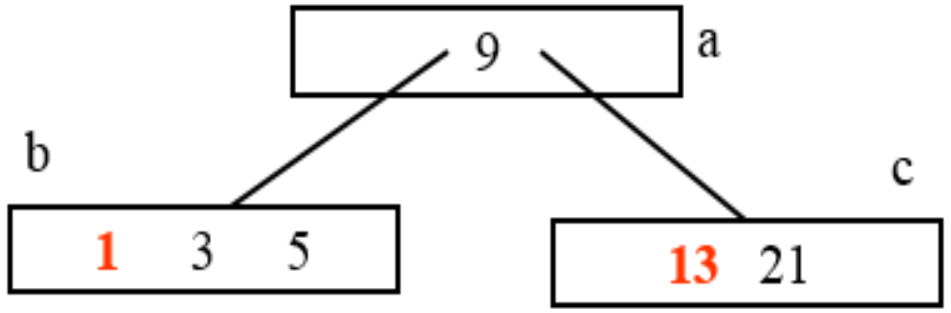


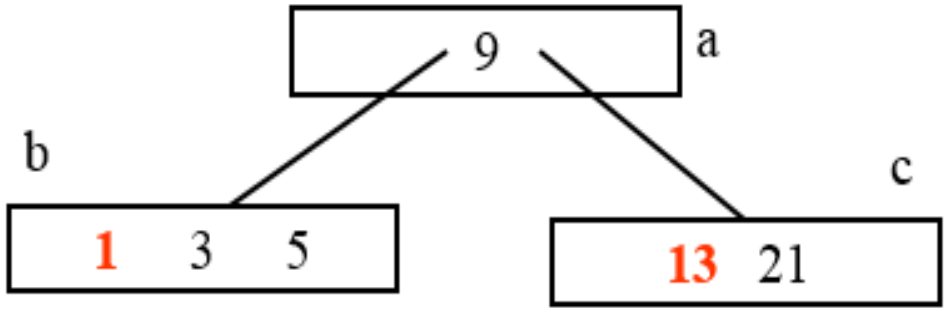
در گره a سرریز رخ داده است
بنابراین این گره باید به دو گره
b و c شکسته شود و میانه
(عنصر سوم) به گره بالایی
فرستاد شود

(گره جدید a ایجاد می شود.).

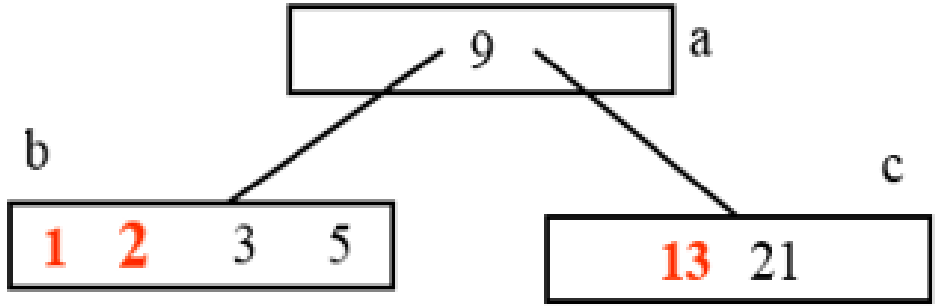


درج اعداد ۱ و ۱۳

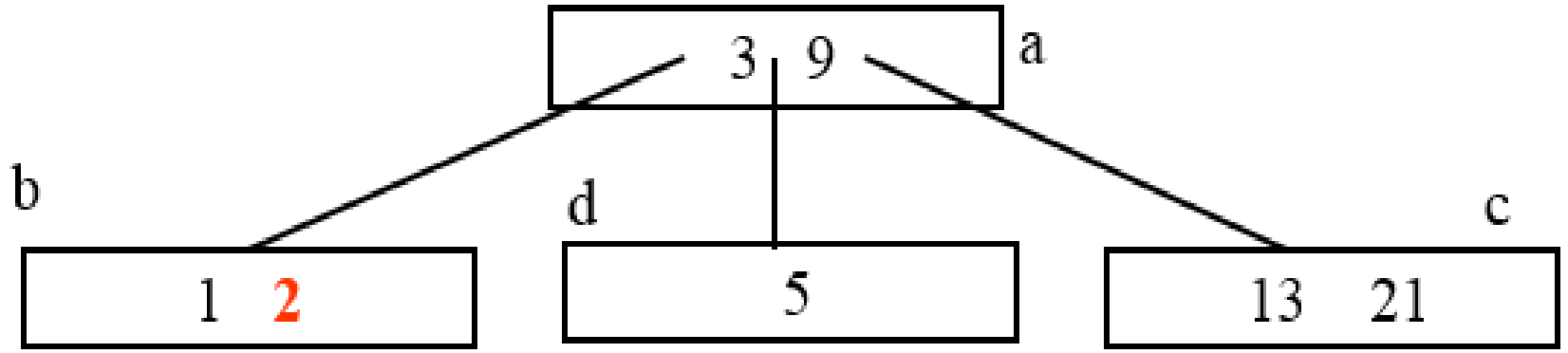


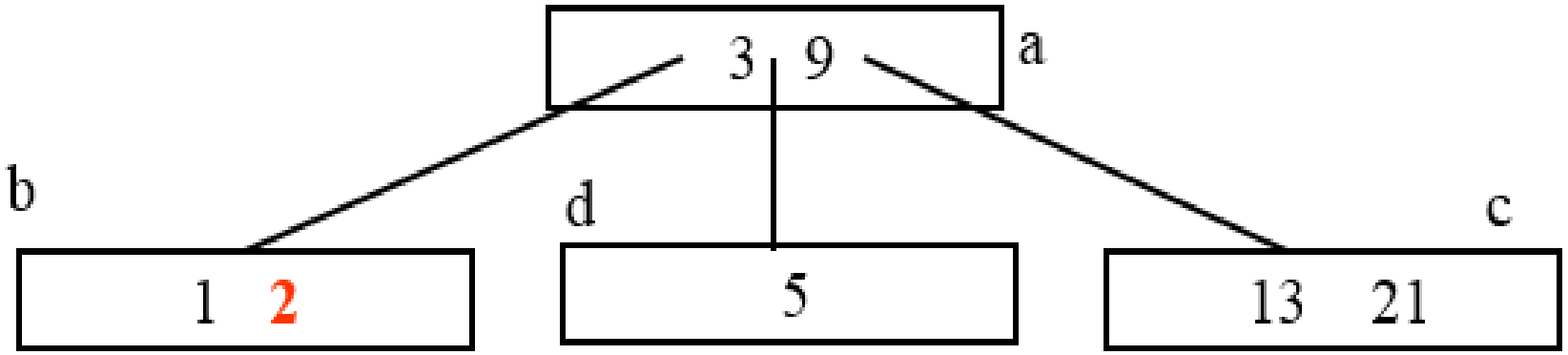


درج ۲

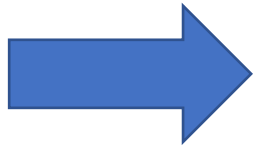


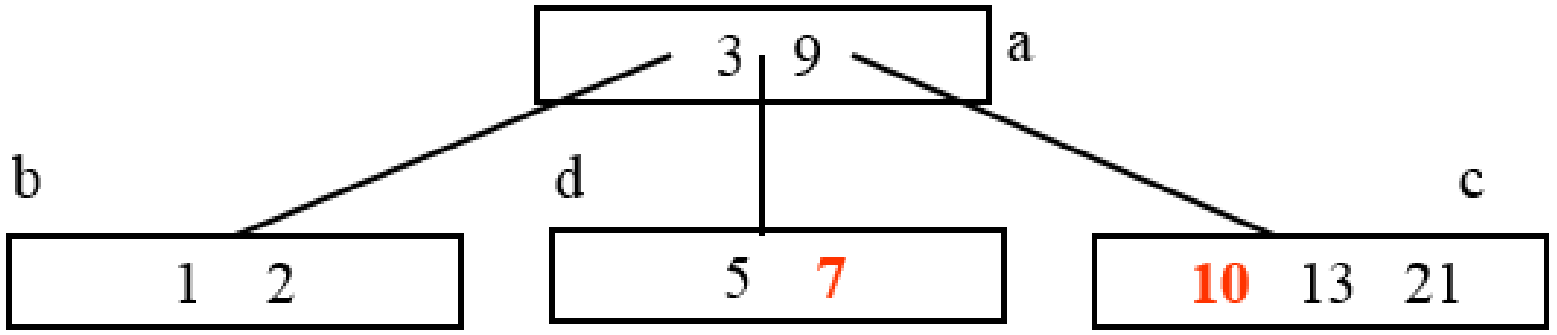
گره b سرریز شده و باید شکسته شود بنابراین کلید ۳ باید به ریشه منتقل شود بقیه به دو گره b و d تقسیم شوند..



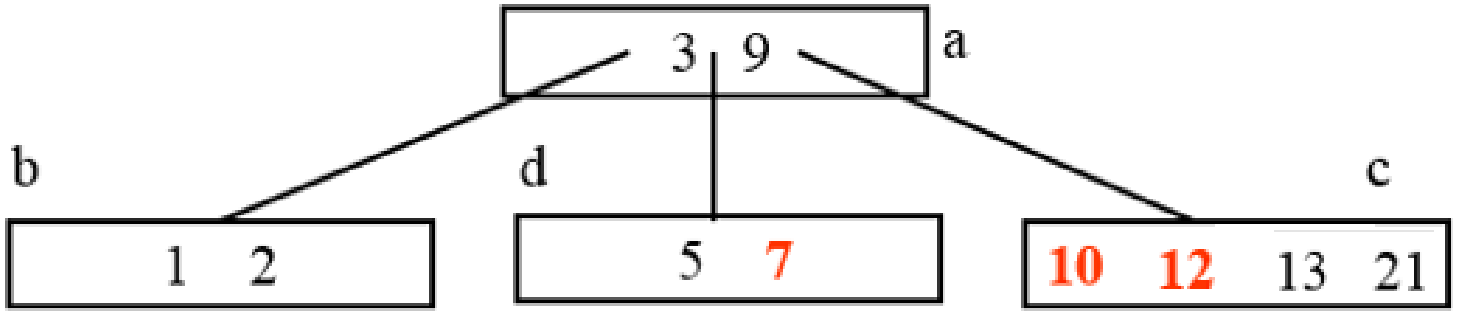


درج اعداد ۷ و ۱۰

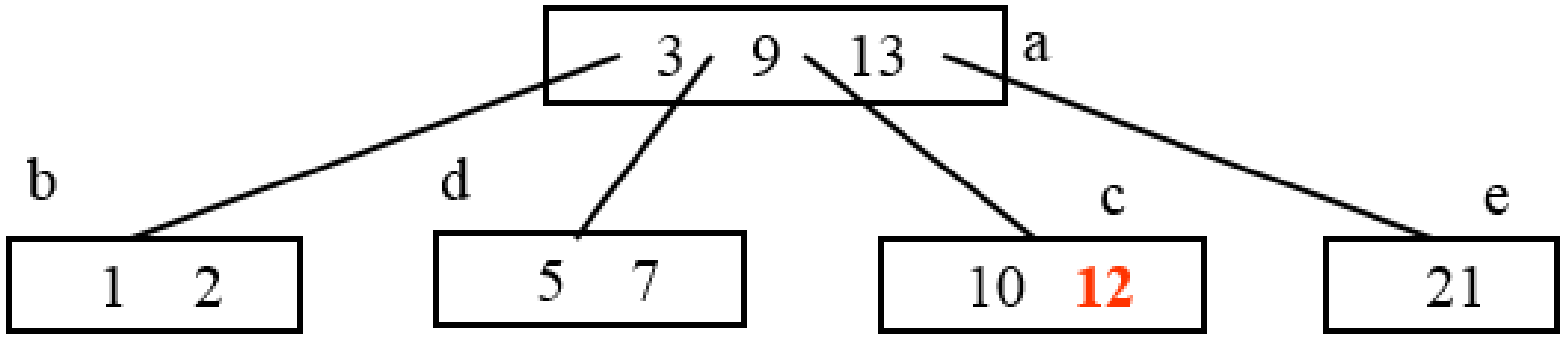


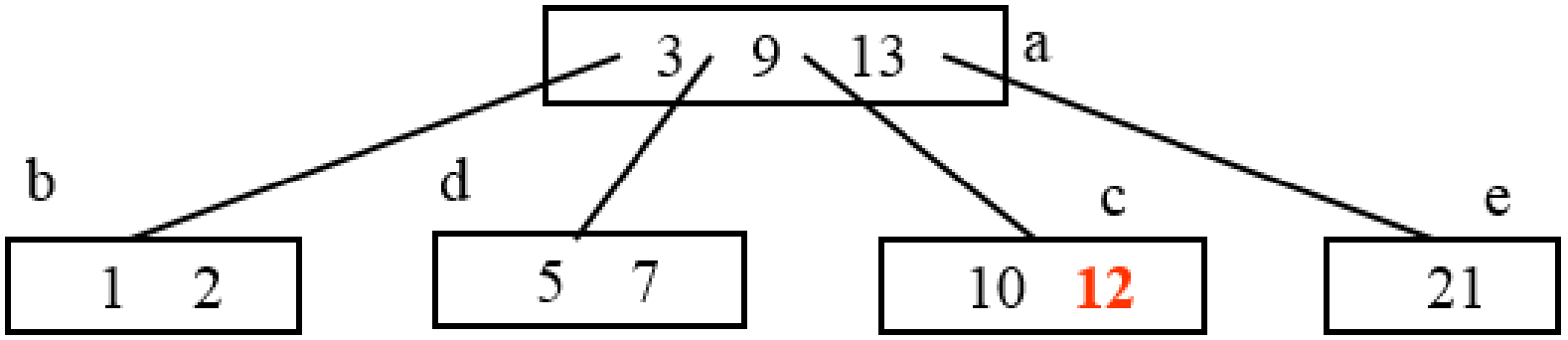


درج عدد ۱۲



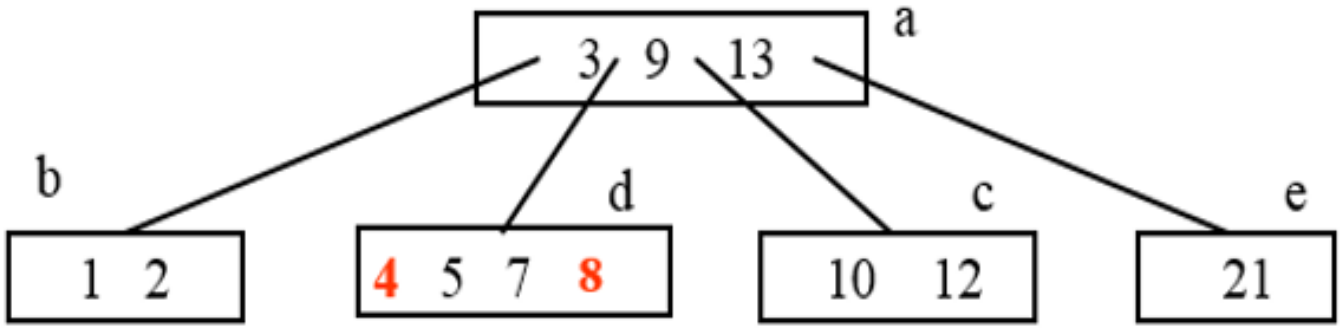
گره c شکسته می شود.



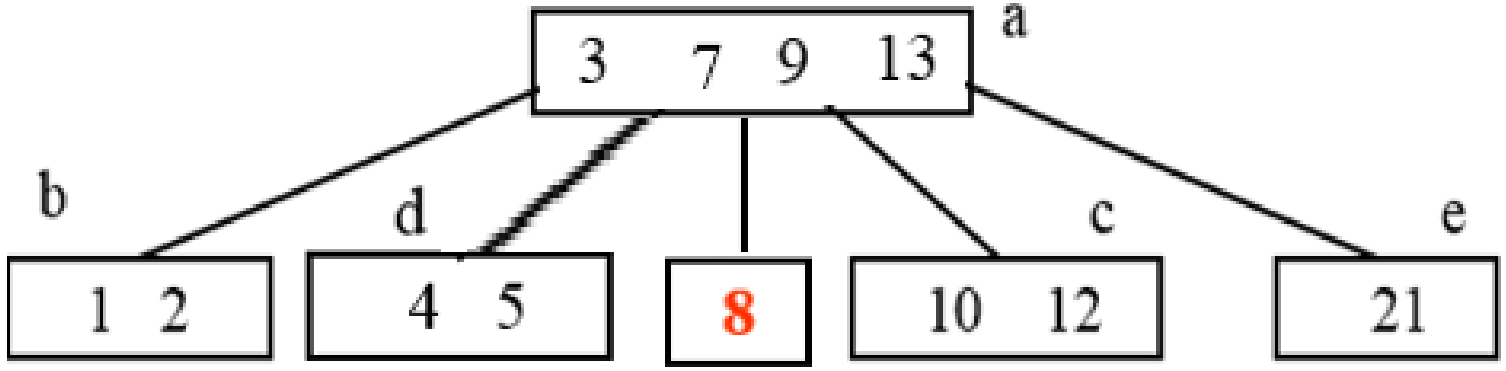


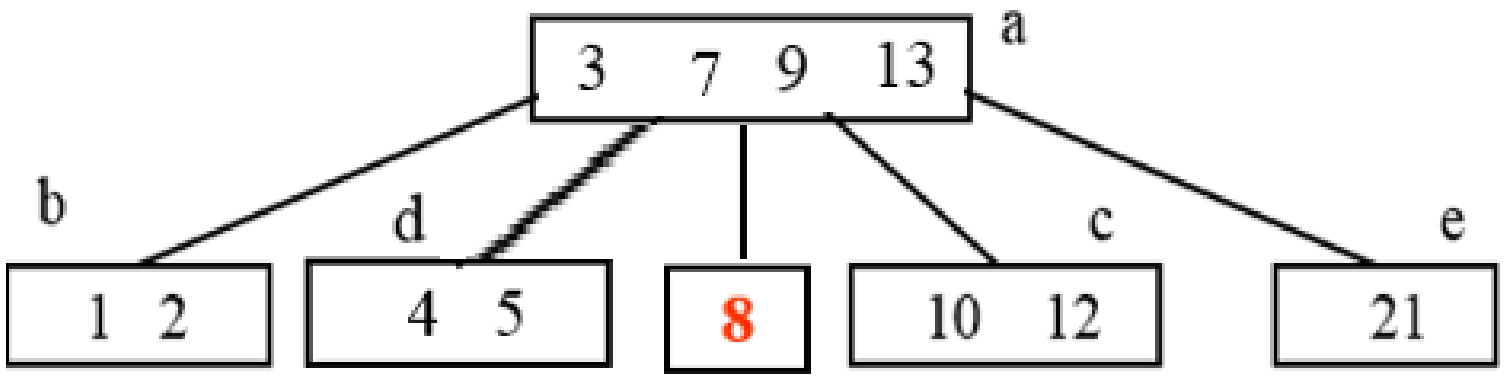
درج عدد ۴

درج عدد ۸

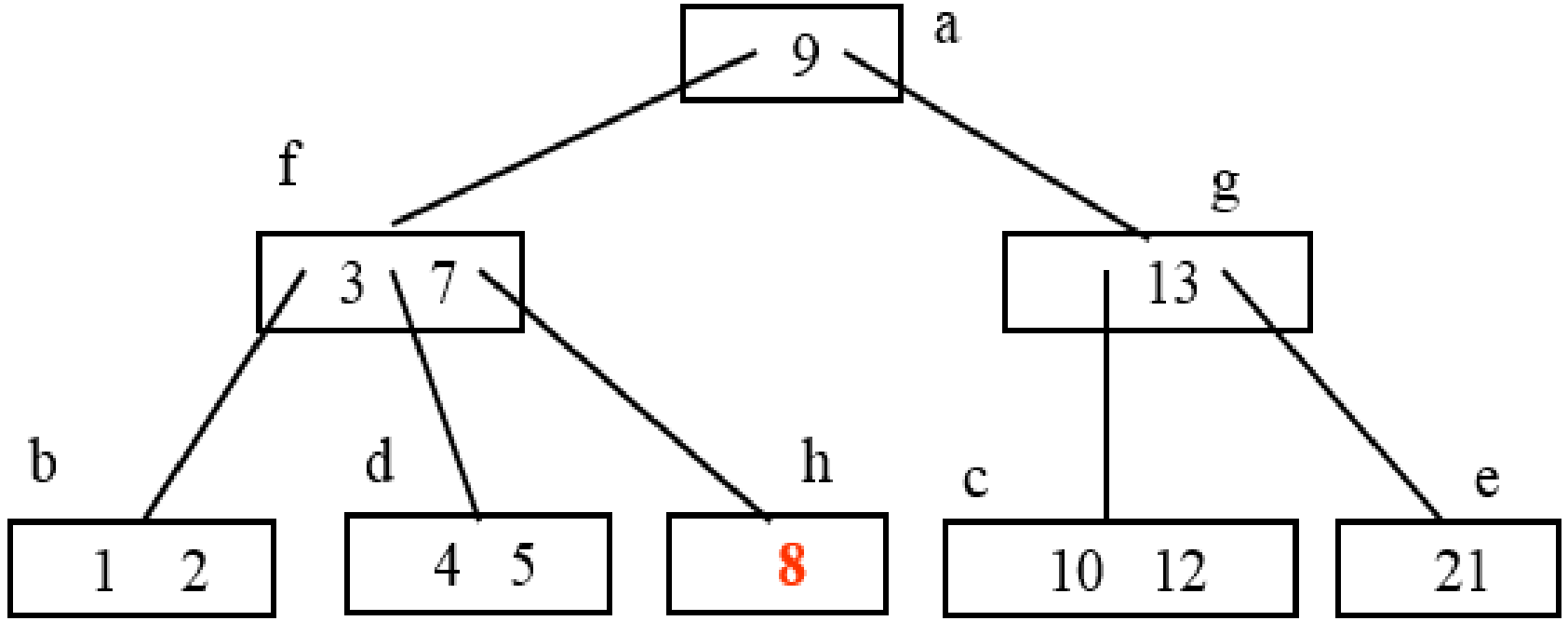


گره d سرریز دارد و به دو گره جدید شکسته می شود و عدد ۷ به گره a فرستاده می شود.





گره a سرریز دارد و به دو گره جدید شکسته می شود و عدد 9 به بالا فرستاده می شود و در نتیجه ارتفاع درخ یک واحد افزایش پیدا می کند.

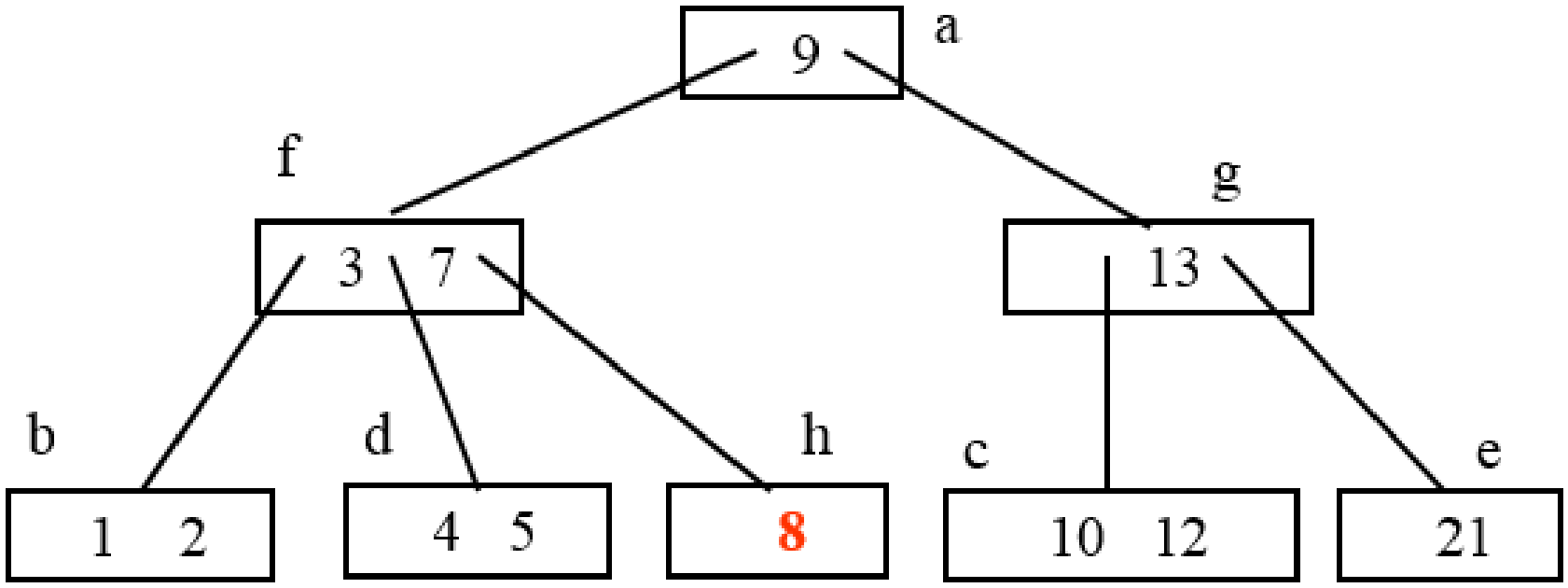


حذف از درخت B (B-tree)

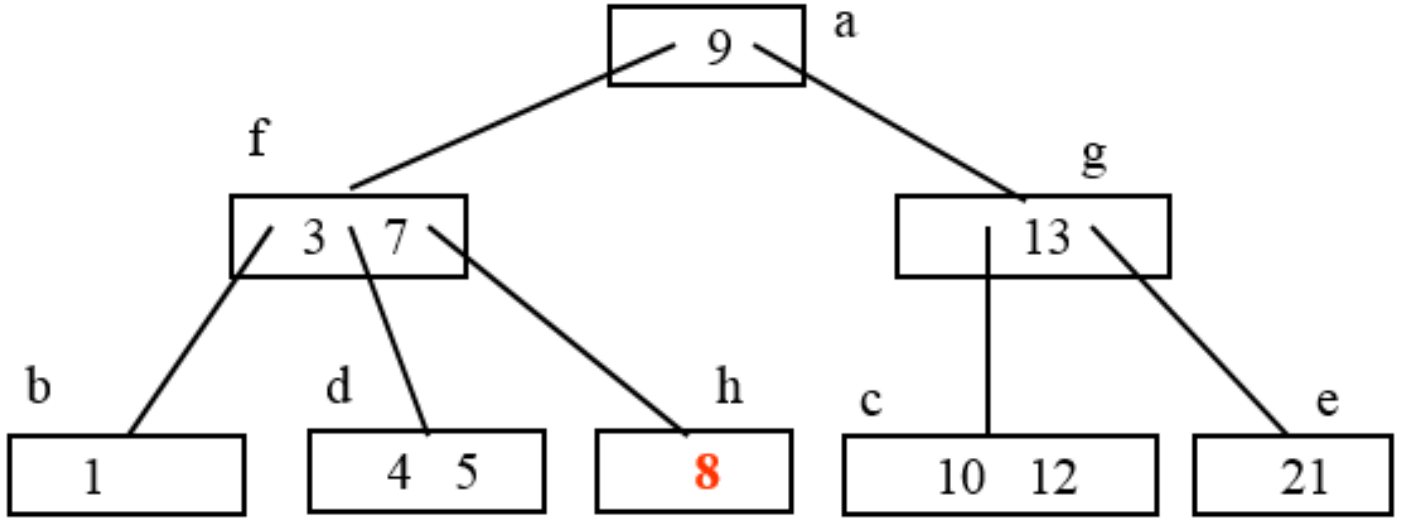
✓ برای حذف x از یک درخت B با درجه t، ابتدا جای آن را پیدا کرده و حذف می کنیم. اگر تعداد کلیدهای گره حاصله کمتر از t-1 باشد (زیرریز: underflow) باید کلیدهایی را از گره های دیگر به این گره بیاوریم.

✓ مثال: درجه درخت

برابر است با t=2

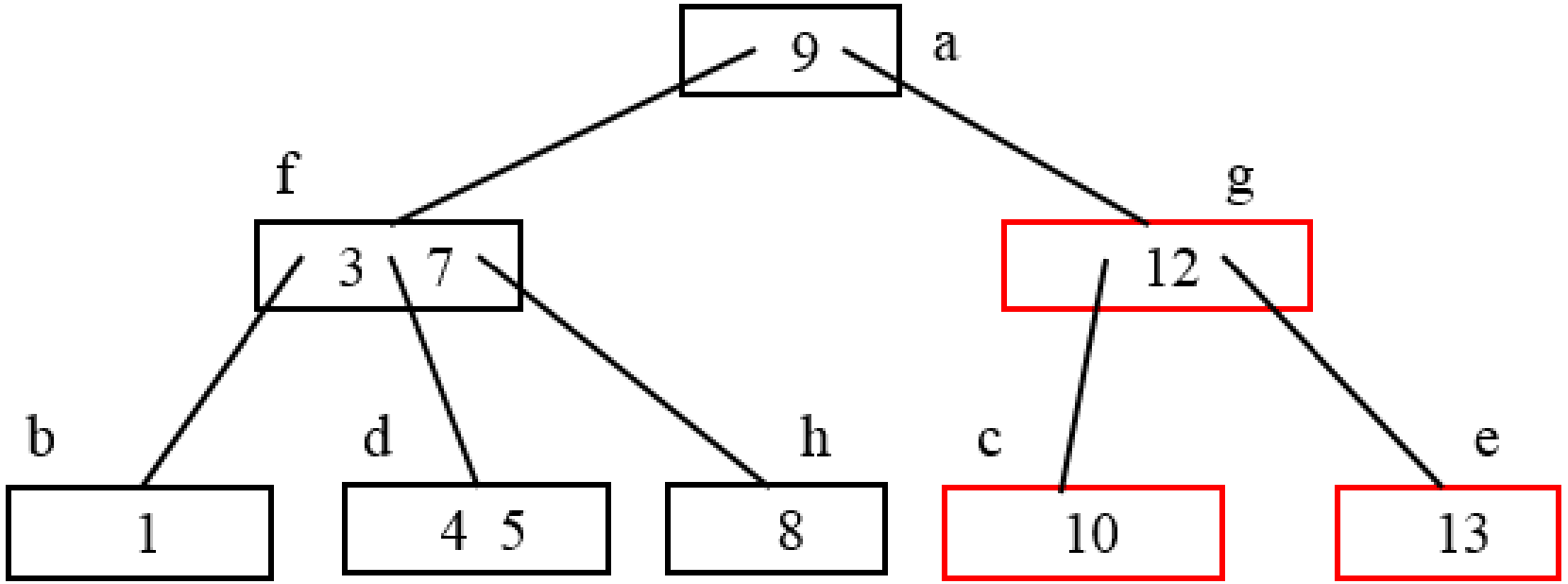


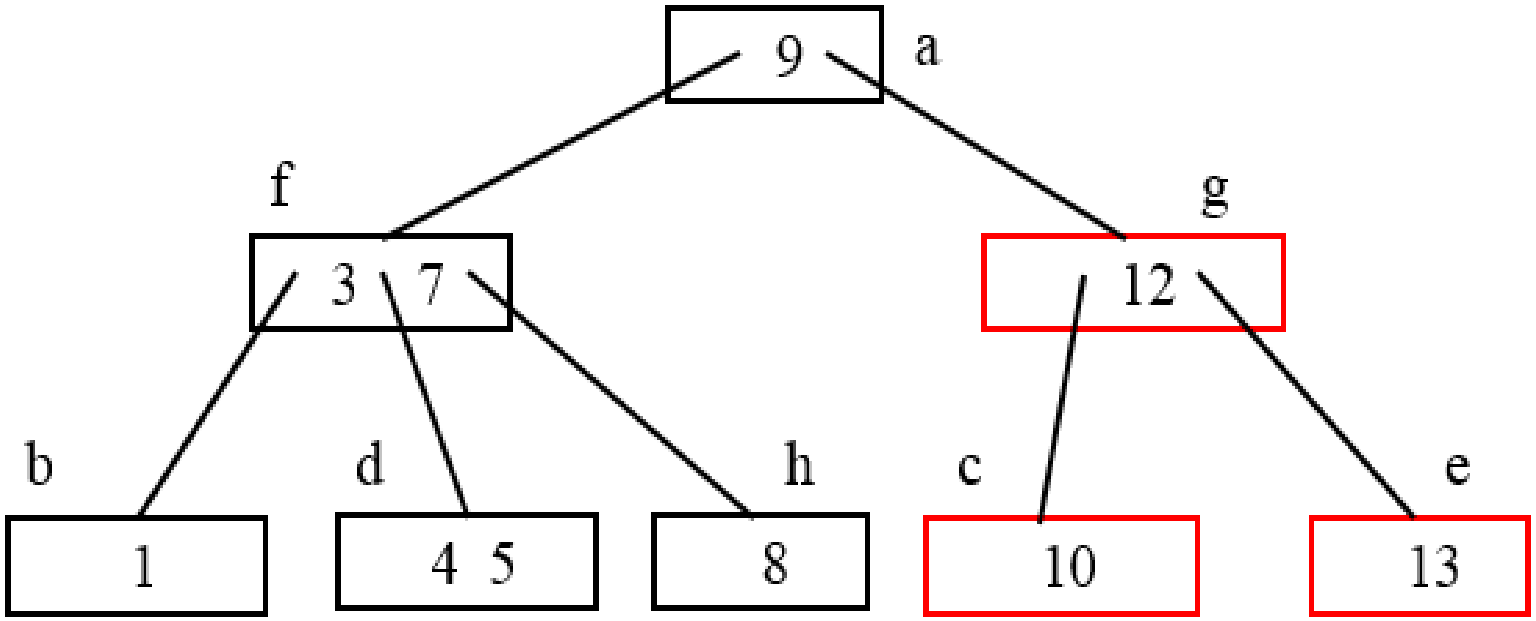
حذف عدد ۲



حذف عدد ۲۱

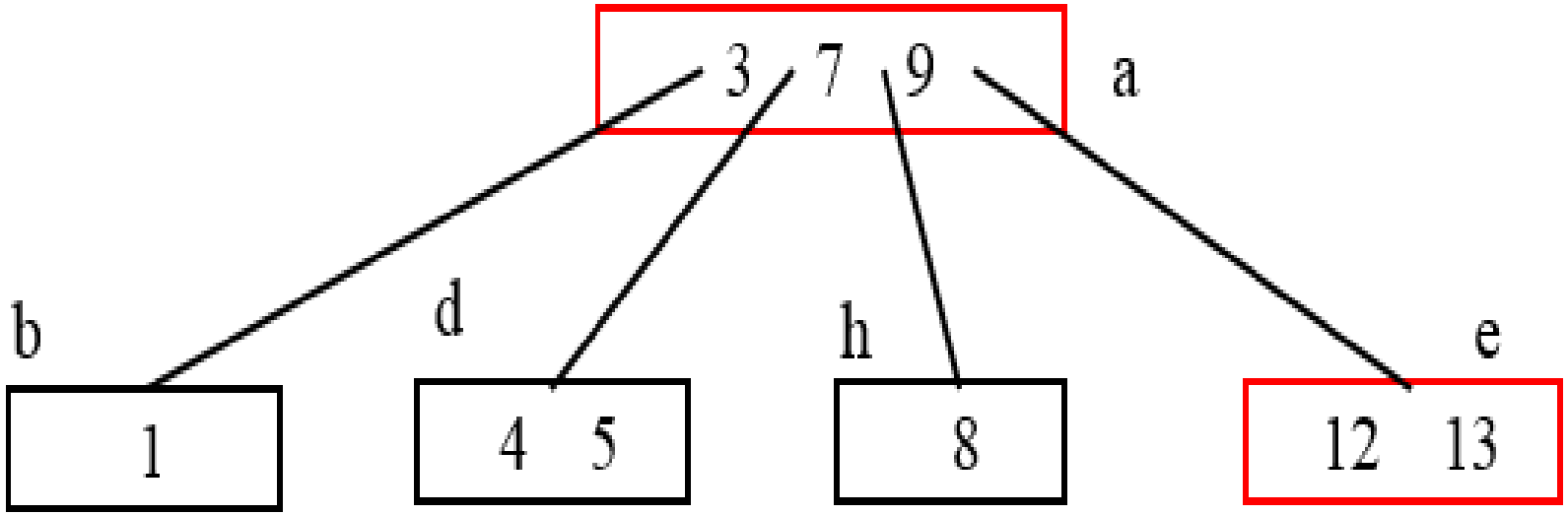
باعث می شود که گره e زیرریز بشود
بنابراین کلیدهای گره های c و g در سه گره c,g,e دوباره توزیع می شود.





حذف عدد ۱۰

باعث می شود که گره c زیرریز بشود بنابراین
والد گره c یعنی g با گره e ترکیب شود و
a و f با هم ترکیب شوند
و ارتفاع درخت یک واحد کم شود.



تمرینات سری ۱

۱- درخت B با درجه t=2 با عناصر زیر ایجاد کنید (مرحله به مرحله)

5, 4, 7, 1, 3, 2, 9, 10, 11, 12, 13, 14

۲- مراحل حذف کلید ۱۴ را از درخت حاصله بنویسید.

۳- مراحل حذف کلید ۱۳ را از درخت حاصله بنویسید.

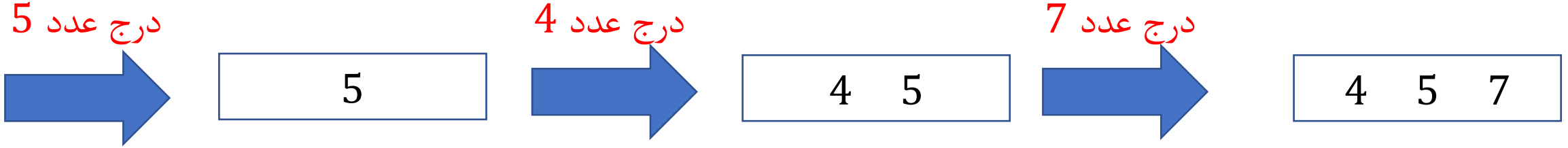
۴- مراحل حذف کلید ۴ را از درخت حاصله بنویسید.

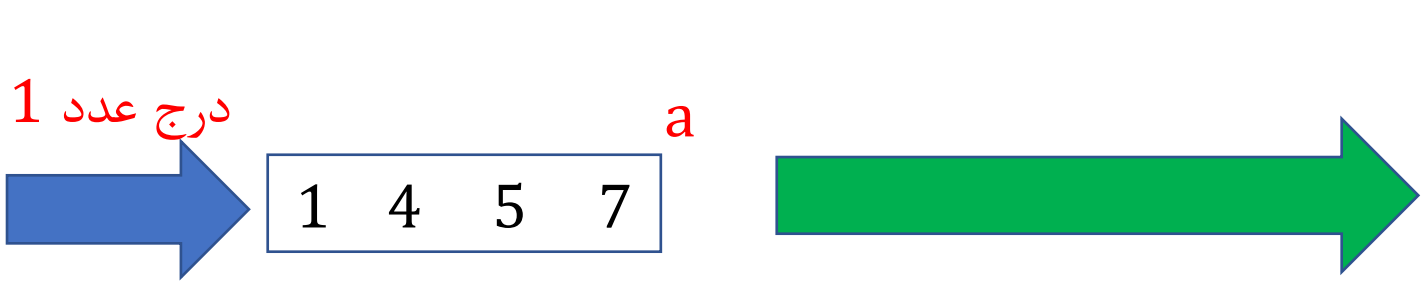
جواب تمرینات سری ۱

۱- درخت B با درجه t=2 با عناصر زیر ایجاد کنید (مرحله به مرحله)

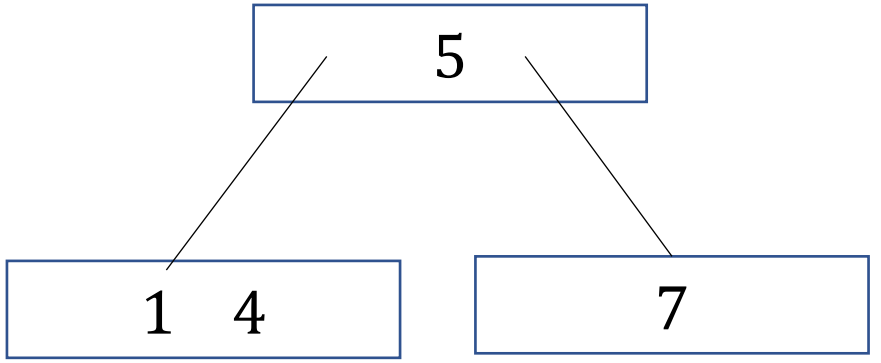
5, 4, 7, 1, 3, 2, 9, 10, 11, 12, 13,14

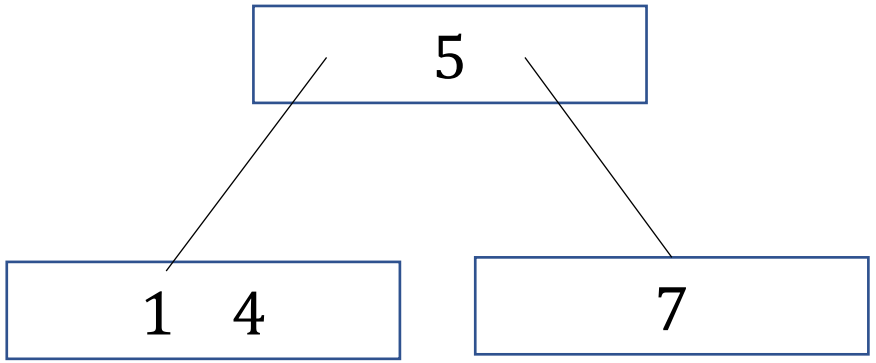
جواب: هر گره حداکثر می تواند 2t-1=3 کلید و 2t=4 فرزند داشته باشد.



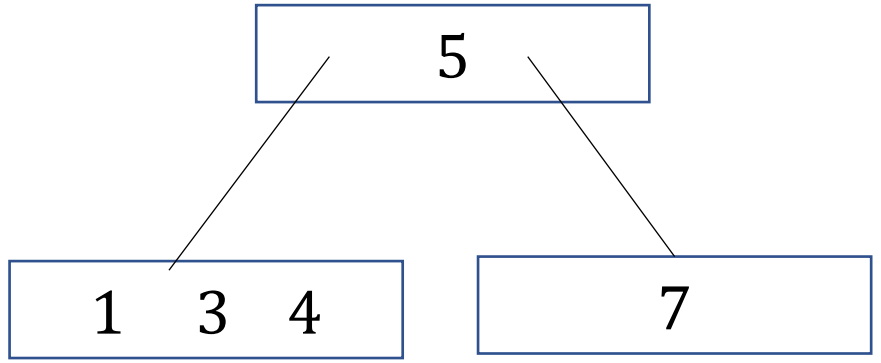


سرریز رخ داده است و باید گره a شکسته شود. عنصر سوم به ریشه منتقل می شود و بقیه به دو گره تقسیم می شوند.

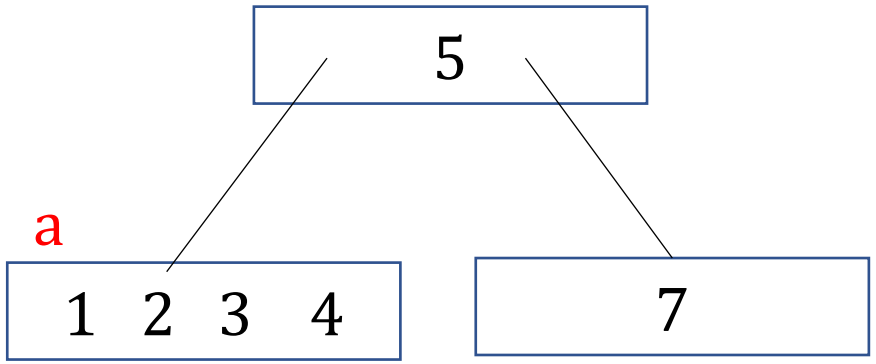




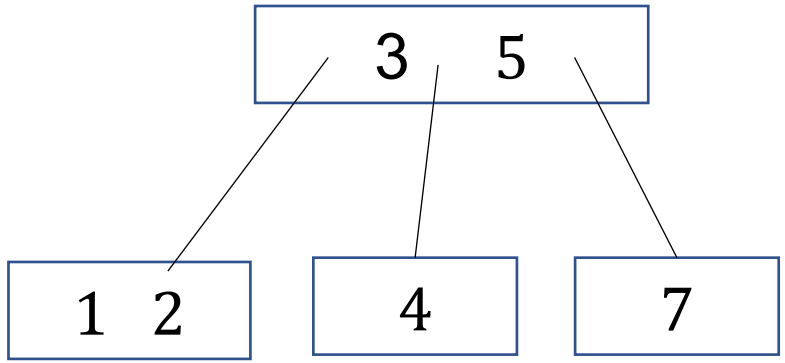
درج عدد 3

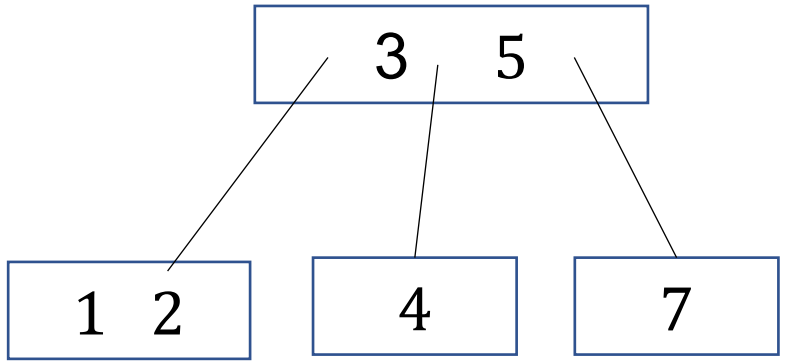


درج عدد 2

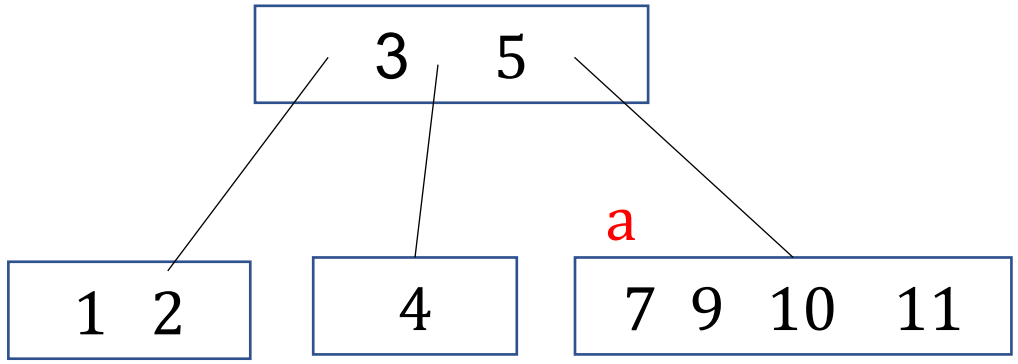


سرریز رخ داده است و باید گره a شکسته شود. عنصر سوم به ریشه منتقل می شود و بقیه به دو گره تقسیم می شوند.

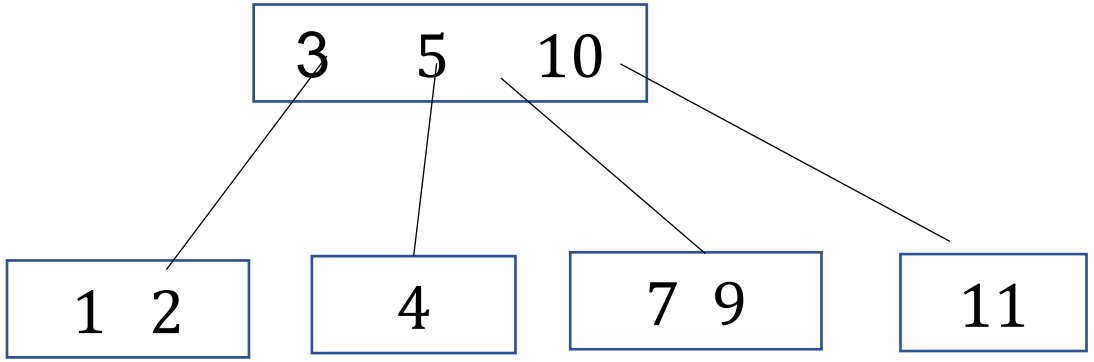


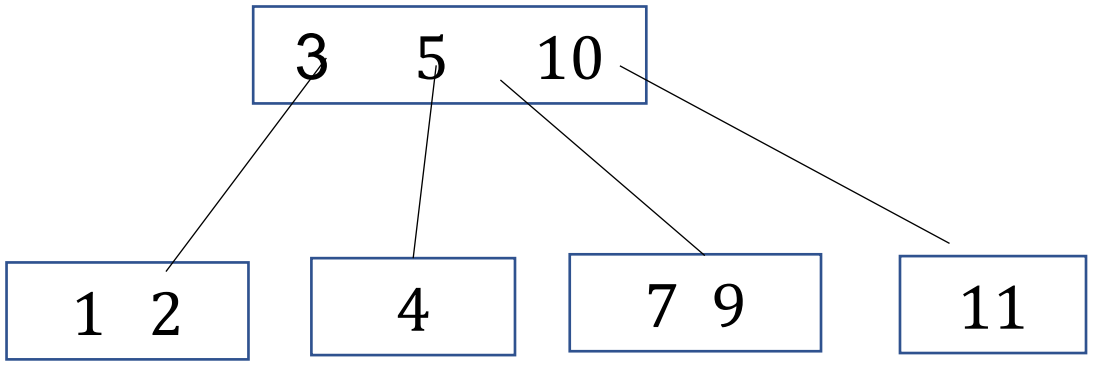


درج اعداد ۹ و ۱۰ و ۱۱

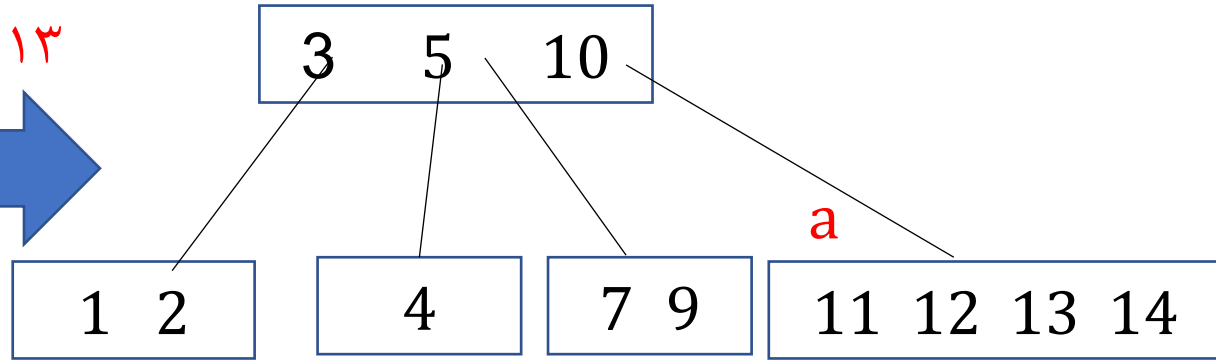


سرریز رخ داده است و باید گره a شکسته شود. عنصر سوم گره a به ریشه منتقل می شود و بقیه به دو گره تقسیم می شوند.

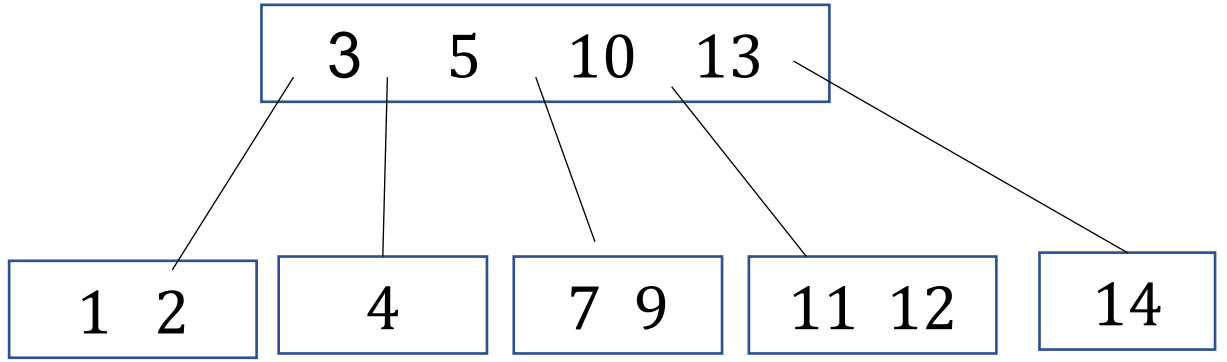


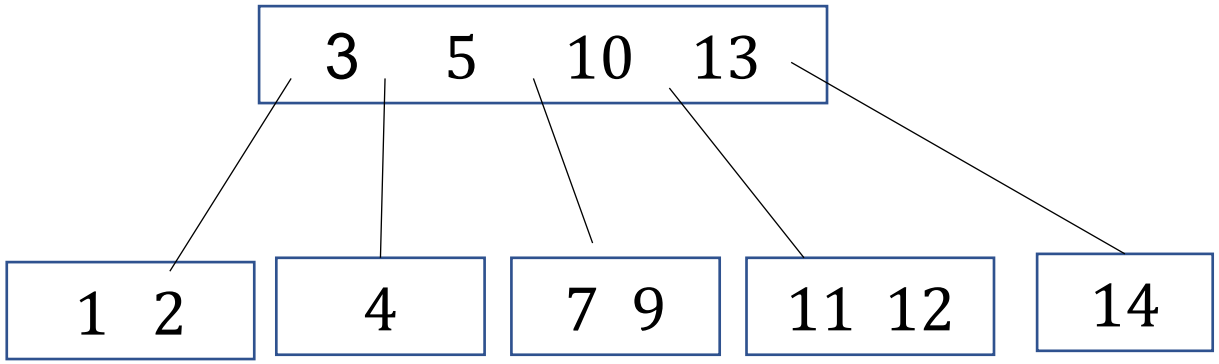


درج عدد ۱۲ و ۱۳ و ۱۴

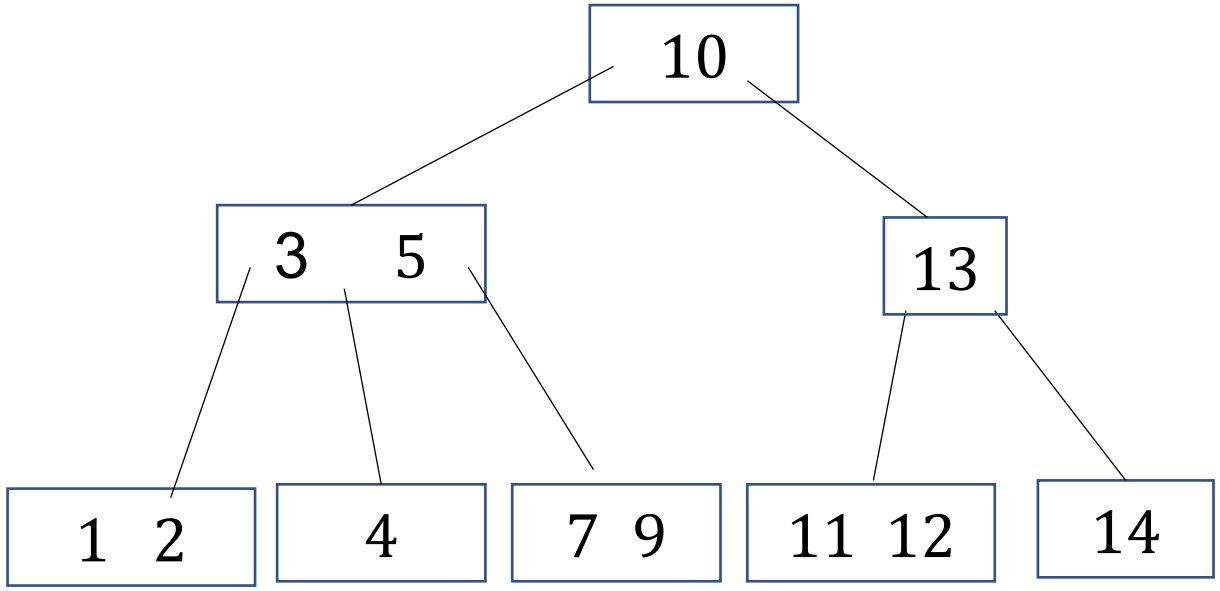


سرریز رخ داده است و باید گره a شکسته شود. عنصر سوم گره a به ریشه منتقل می شود و بقیه به دو گره تقسیم می شوند.

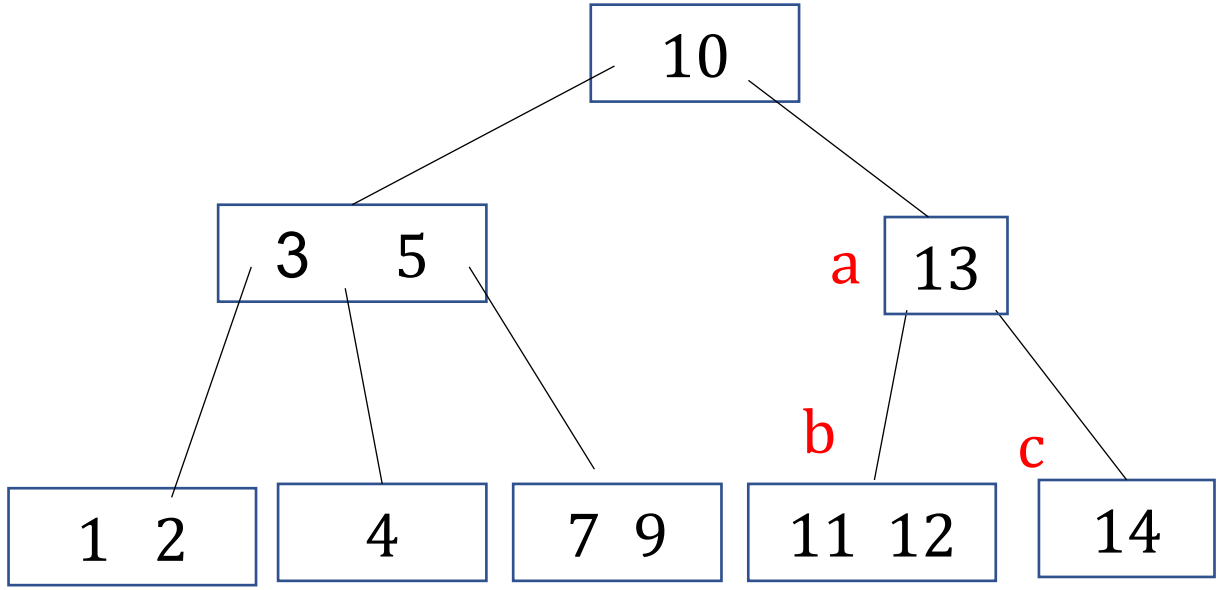




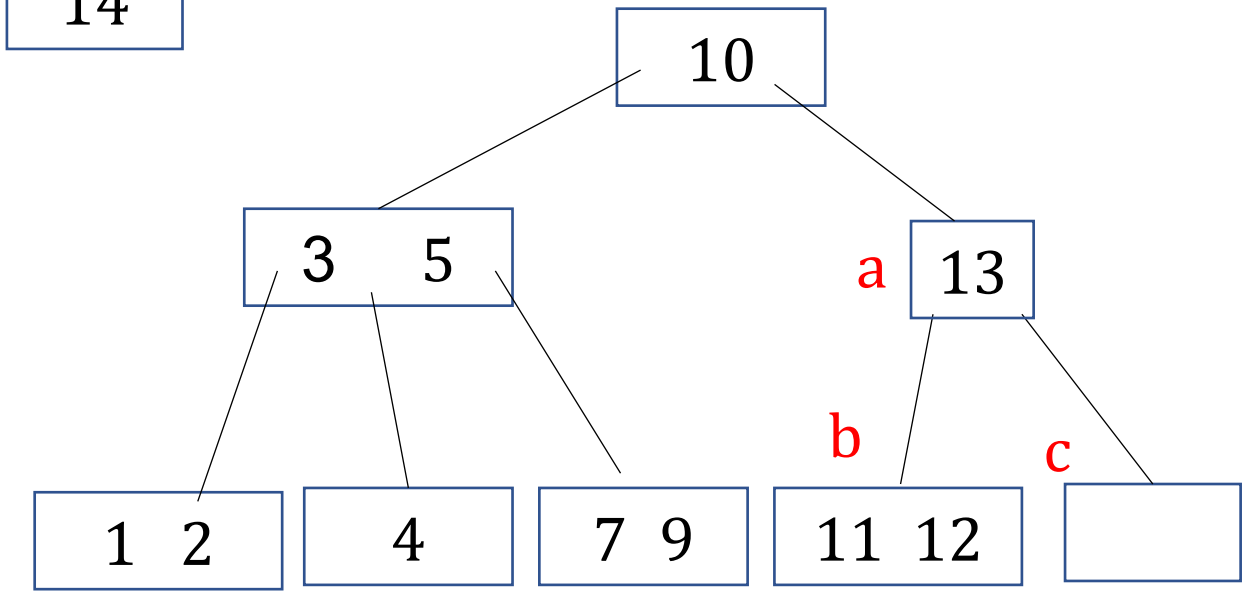
در ریشه سرریز رخ داده است و باید شکسته شود. عنصر سوم یعنی ۱۰ به بالا منتقل می شود بقیه به دو گره تقسیم می شوند ضمنا ارتفاع درخت یک واحد افزایش پیدا می کند.

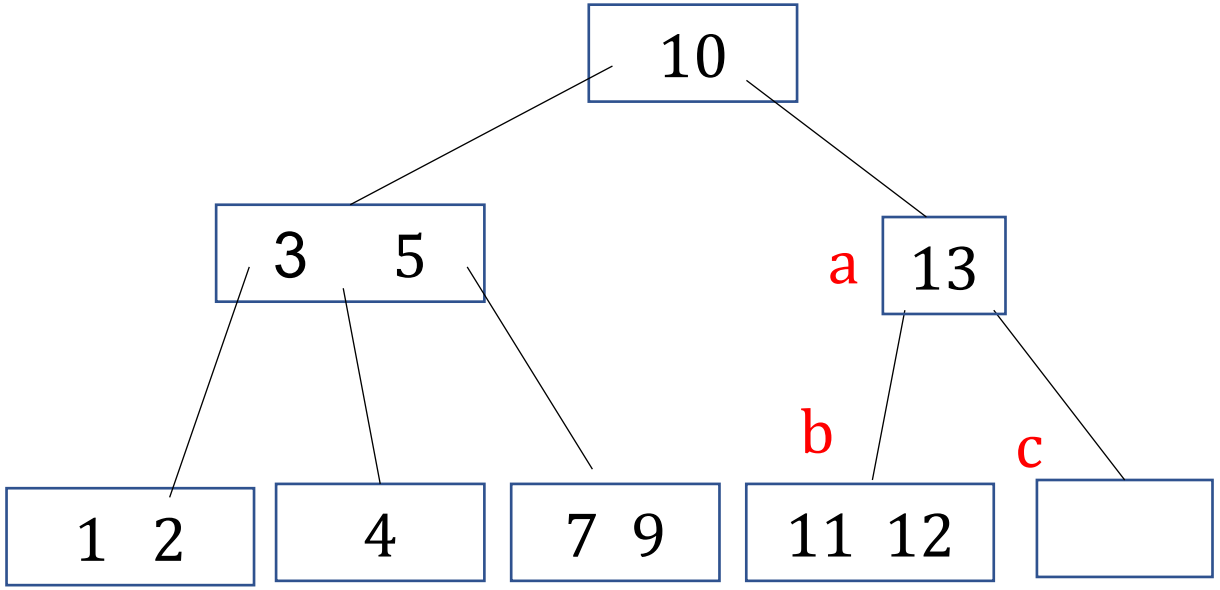


۲- مراحل حذف کلید ۱۴ را از درخت حاصله بنویسید.

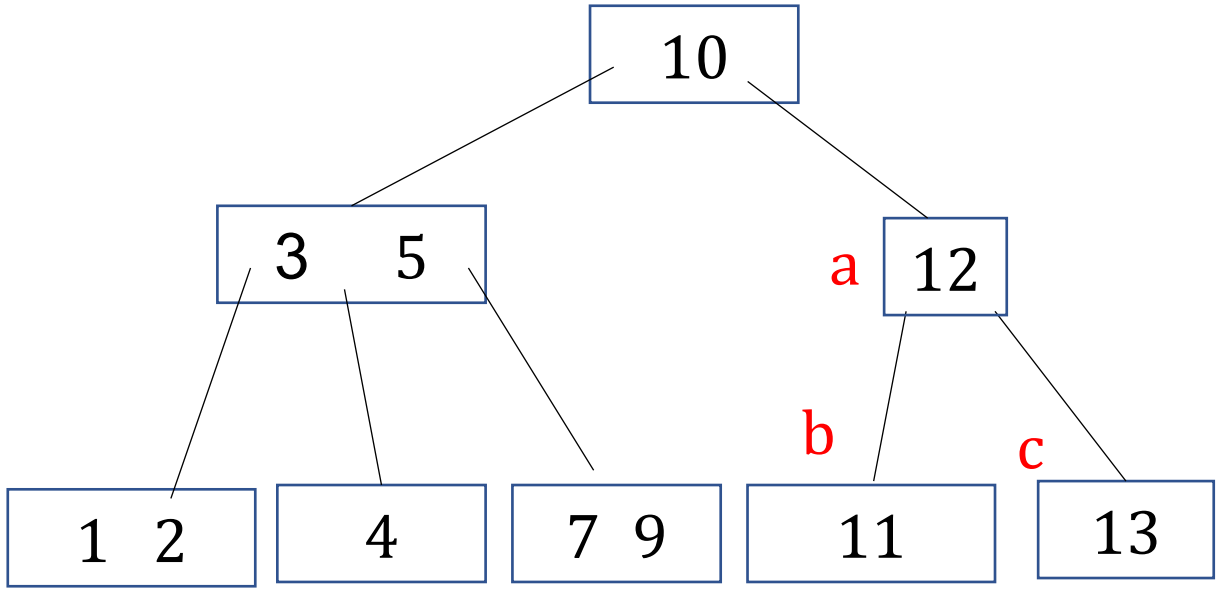


حذف عدد ۱۴

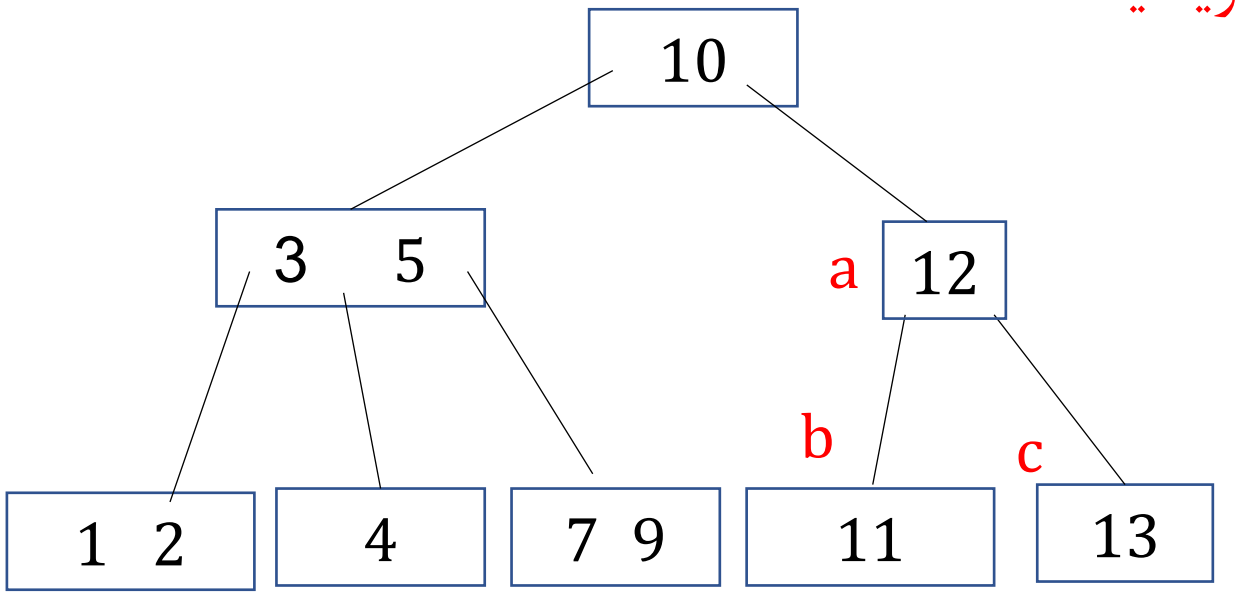




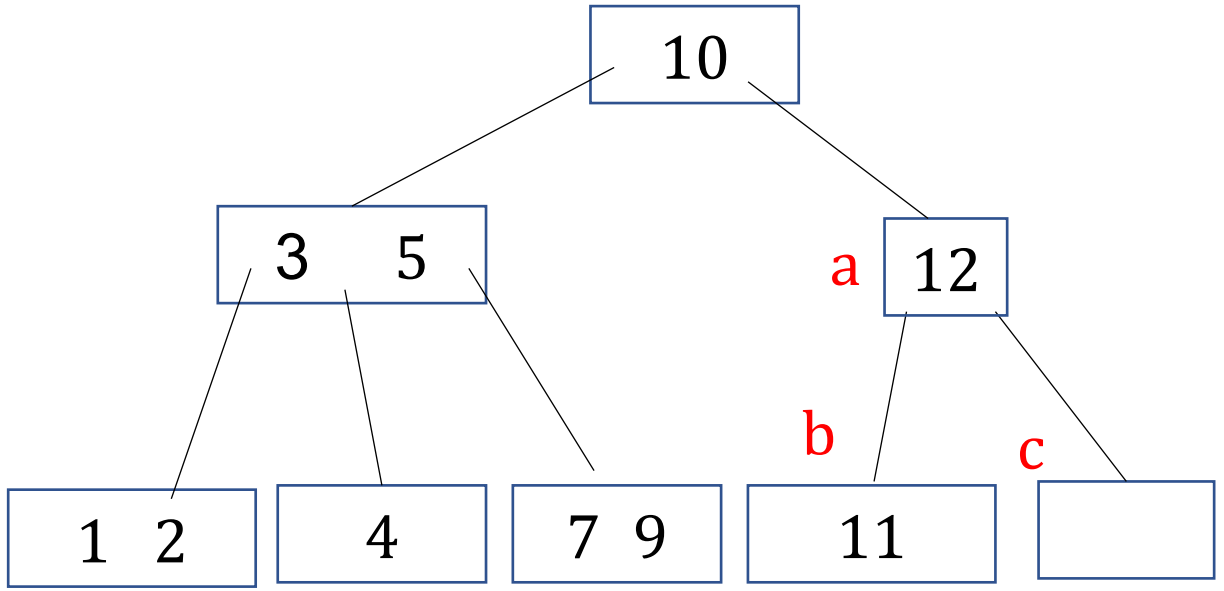
گره c زیرریز شده است بنابراین
والد گره c یعنی a با گره b ترکیب شده و
و در سه گره جدید توزیع می شود.

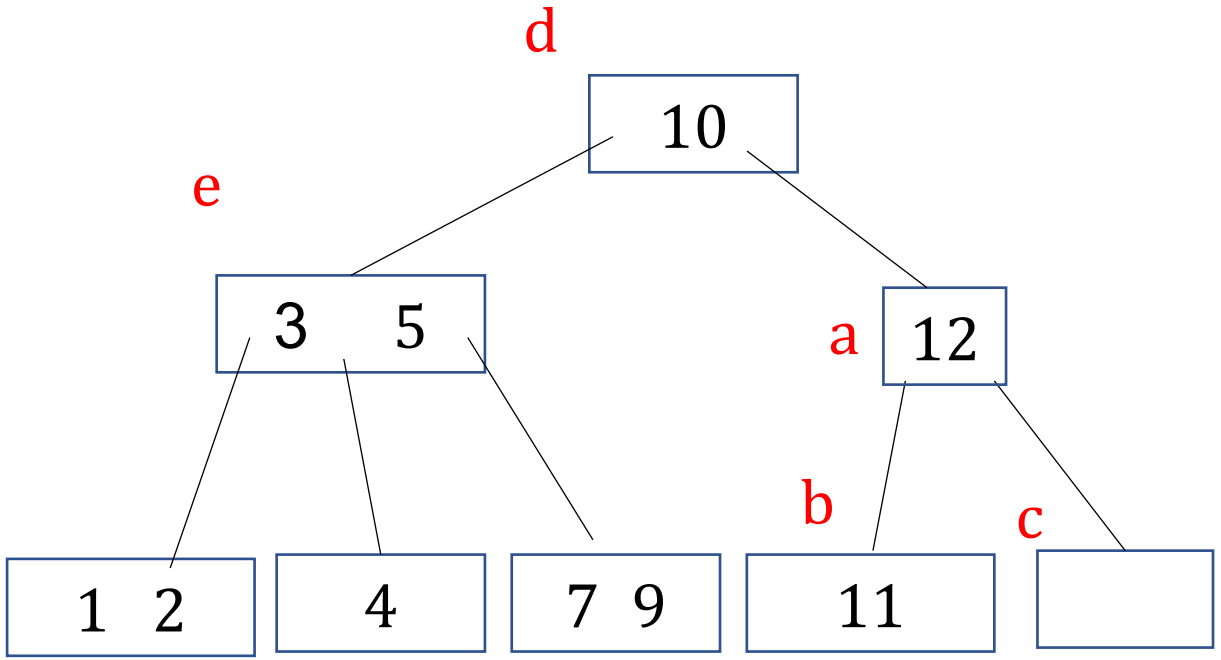


۳- مراحل حذف کلید ۱۳ را از درخت حاصله بنویسید.

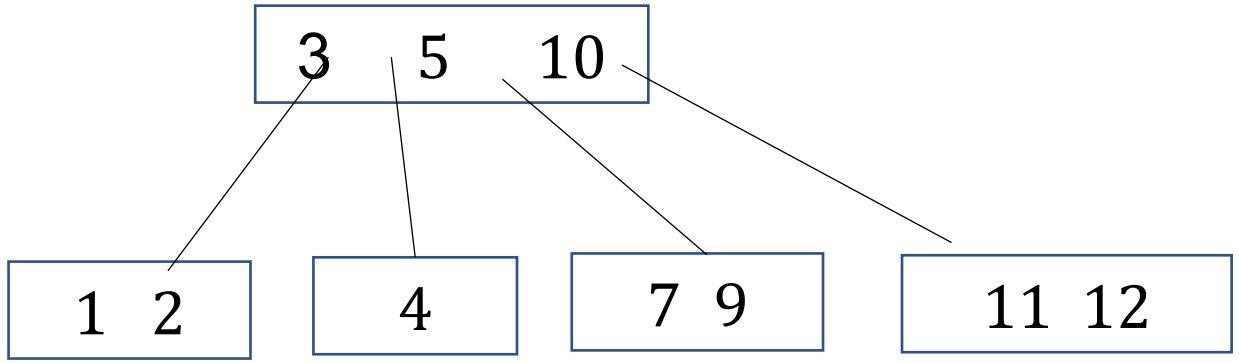


حذف عدد ۱۳

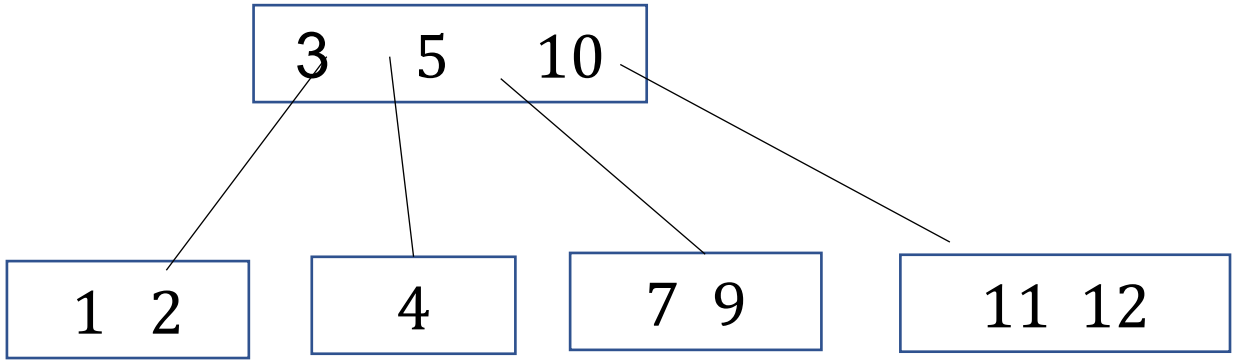




گره c زیرریز شده است بنابراین والد گره c یعنی a با گره b ترکیب شده و d با e ترکیب می شوند و ارتفاع درخت یک واحد کم می شود.

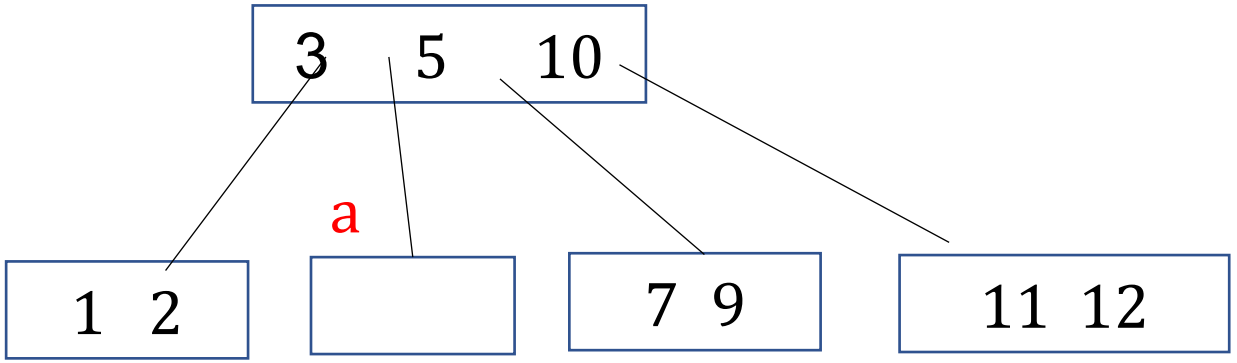


۴- مراحل حذف کلید ۴ را از درخت حاصله بنویسید.

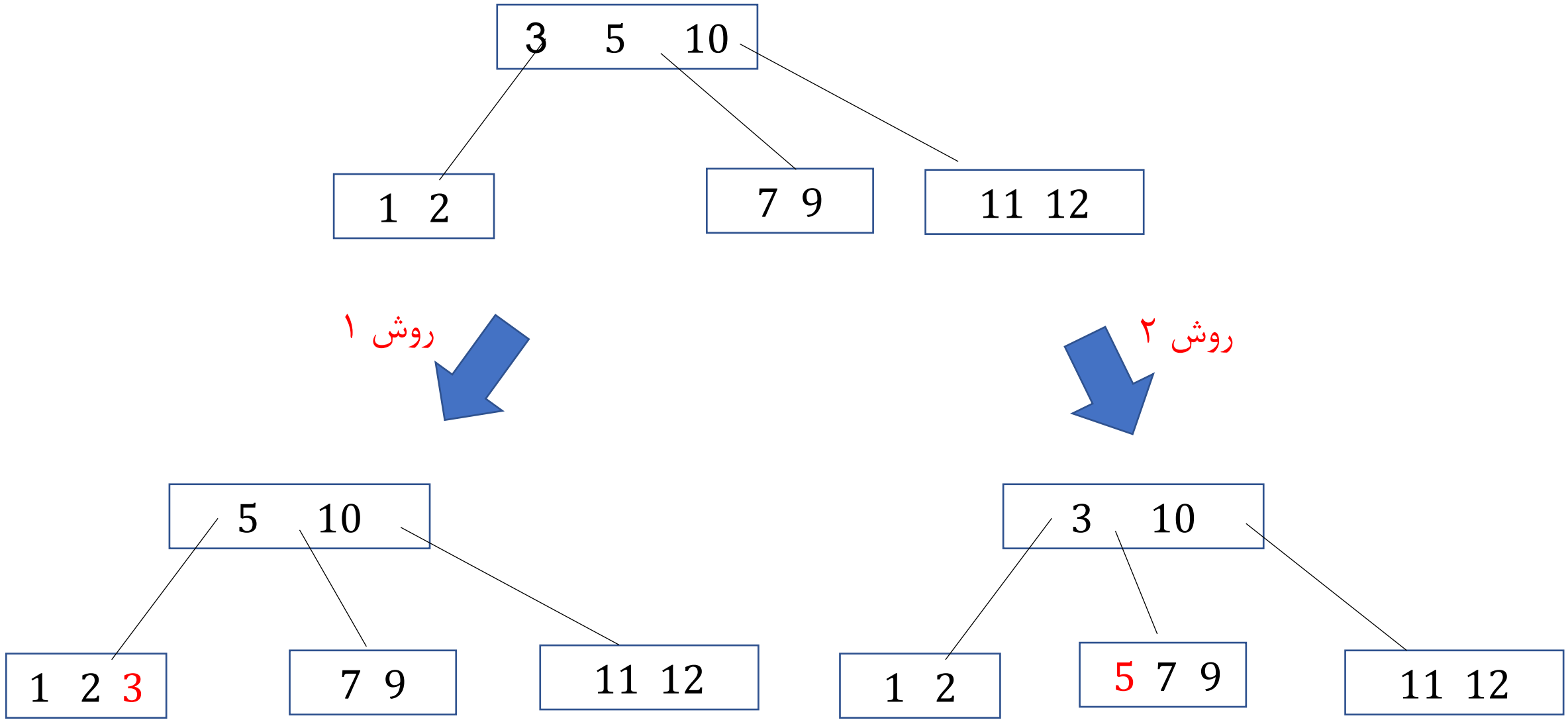


حذف عدد ۴

.



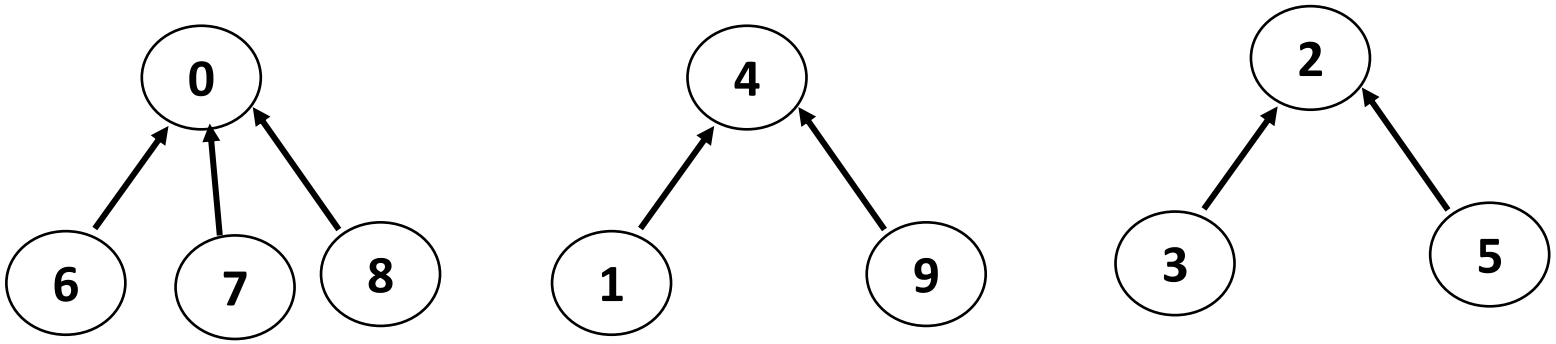
گره a زیرریز شده است


3 5 10
1 2
7 9
11 12
روش ۱
روش ۲
5 10
1 2 3
7 9
11 12
3 10
1 2
5 7 9
11 12

مجموعه های مجزا (Disjoint Sets)

- ✓ تعریف: چند مجموعه را مجزا از هم گویند اگر اشتراکی نداشته باشند.

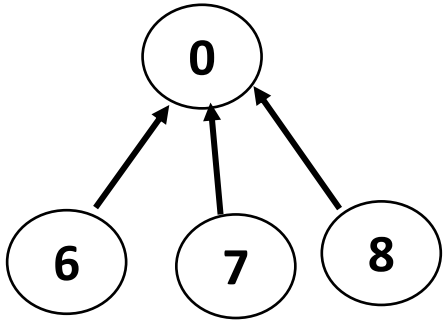
- ✓ یک روش برای نگهداری مجموعه ها استفاده از جنگل می باشد.
- ✓ بعنوان مثال: شکل زیر نگهداری سه مجموعه $S1=\{0,6,7,8\}$, $S2=\{1,4,9\}$, $S3=\{2,3,5\}$ را نشان می دهد. مهم نیست که چه عنصری از مجموعه در ریشه درخت متناظر قرار می گیرد.
- ✓ برخلاف درخت معمولی، در اینجا از فرزندان به والد اشاره گر داریم.



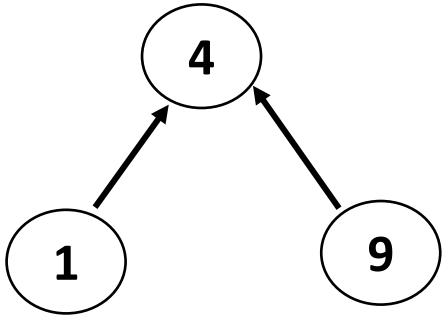
اعمال روی مجموعه های مجزا

۱- union (اجتماع): اجتماع دو مجموعه را نشان می دهد. برای محاسبه اجتماع دو مجموعه S1 و S2 کافی است که ریشه درخت S1 را فرزند ریشه درخت S2 قرار دهیم.

بعنوان مثال:

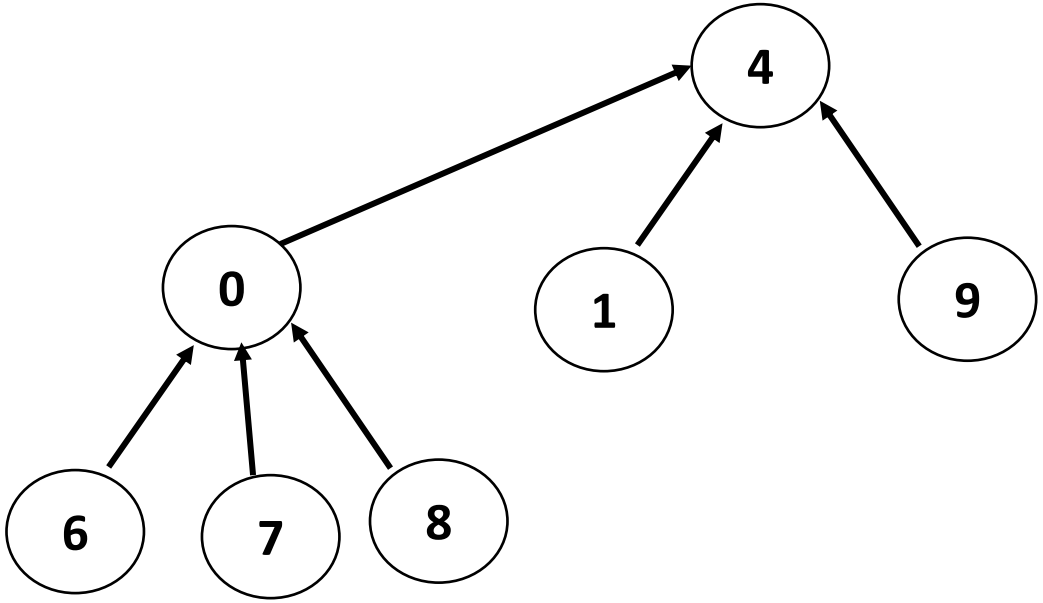


S1



S2

union(S1, S2)



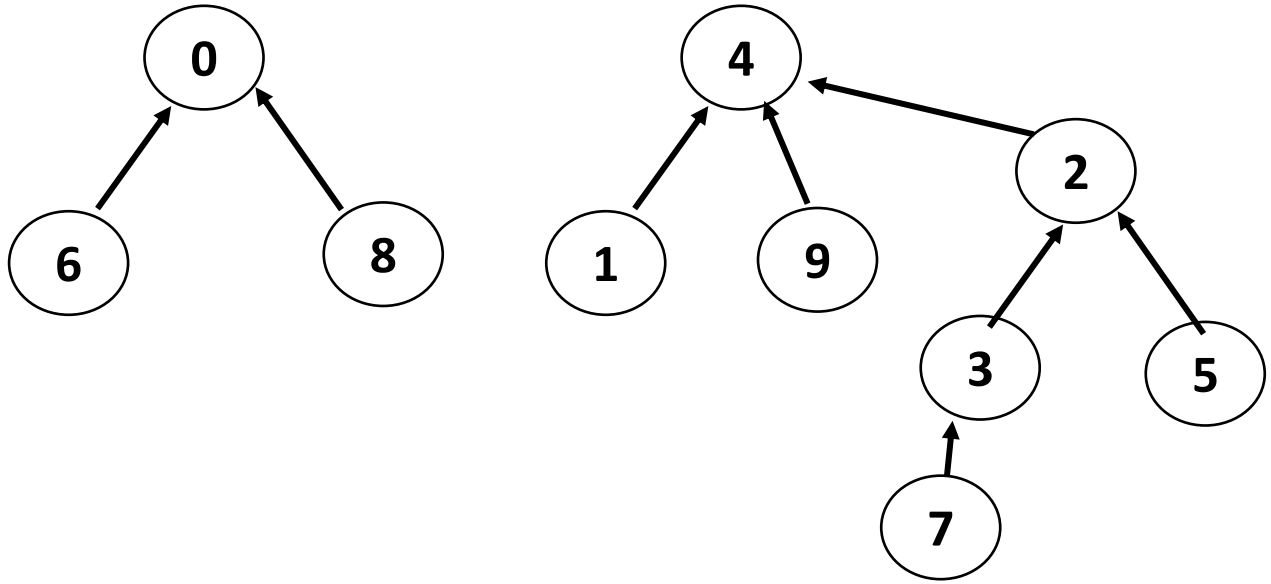
۲- find(x) (جستجو): مجموعه ای که x عضو آن است را پیدا می کند.

بعنوان مثال: ۳ عضو مجموعه S3 است.

نکته: در ادامه، برای راحتی کار، هر مجموعه را با عدد موجود در ریشه آن نامگذاری می کنیم.

نکته: از آنجا که عناصر مجموعه n عضوی از اعداد 0 تا n-1 شماره گذاری می شوند بنابراین می توانیم مجموعه ها را در یک آرایه با نام parent نگهداری کنیم بطوریکه والد عضو i را در parent[i] ذخیره می کنیم. والد ریشه را عدد -1 قرار می دهیم.

بعنوان مثال: دو مجموعه 0 و 4 را بصورت زیر داریم، آن را در آرایه parent ذخیره کنید.



جواب:

	0	1	2	3	4	5	6	7	8	9
parent	-1	4	4	2	-1	2	0	3	0	4

سوال: find(7)=? و به چند مقایسه نیاز دارد؟ ابتدا به parent[7] مراجعه می کنیم. بعد به parent[3] مراجعه کرده و سپس به parent[2] مراجعه کرده و در نهایت به parent[4] مراجعه می کنیم به عدد 1- می رسیم یعنی به ریشه رسیده ایم پس find(7)=4 و ۴ مقایسه لازم داشت.

یک مساله روی مجموعه های مجزا

مساله: فرض کنیم عناصر صفر تا n-1 را در n مجموعه مجزا از هم که هر کدام شامل فقط یک عضو هست بصورت زیر داریم:

اگر n-1 عمل اجتماع (union) و جستجو (find) را بصورت زیر انجام دهیم کل زمان مورد نیاز برای اعمال اجتماع و جستجو را جداگانه محاسبه کنید.

union(0,1) , find(0)

union(1,2) , find(0)

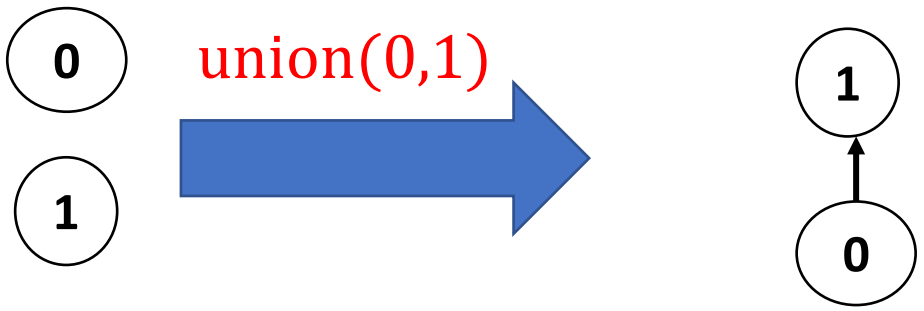
union(2,3) , find(0)

.

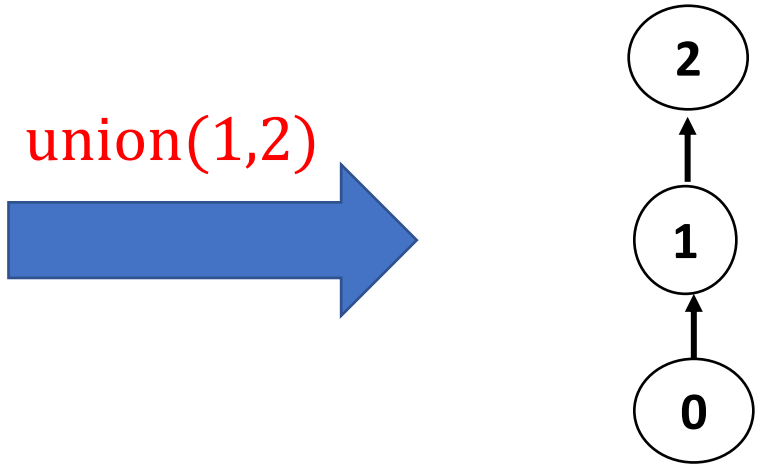
.

union(n-2,n-1) , find(0)

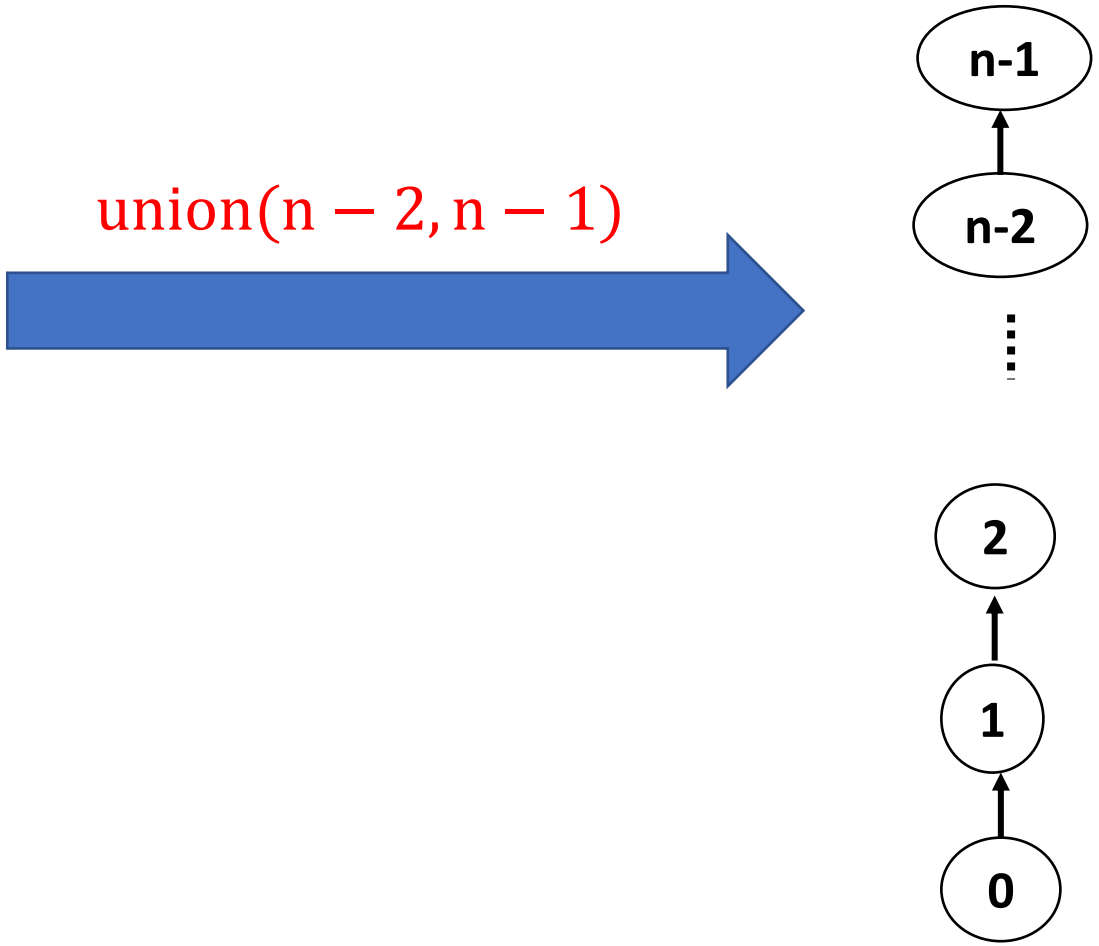
جواب:



عمل اجتماع به زمان ثابت نیاز دارد
عمل find(0) به دو مقایسه نیاز دارد.



عمل اجتماع به زمان ثابت نیاز دارد
عمل find(0) به سه مقایسه نیاز دارد.



عمل اجتماع به زمان ثابت نیاز دارد
عمل $find(0)$ به n مقایسه نیاز دارد.

$T_{union}=1+1+...+1=O(n)$

$T_{find}=2+3+...+n=O(n^2)$

البته می توان با اِعمال قوانینی بر روی عمل اجتماع، ارتفاع درخت حاصل شده را بهینه کرد و در نتیجه زمان های T_{union} و T_{find} را کم کرد.

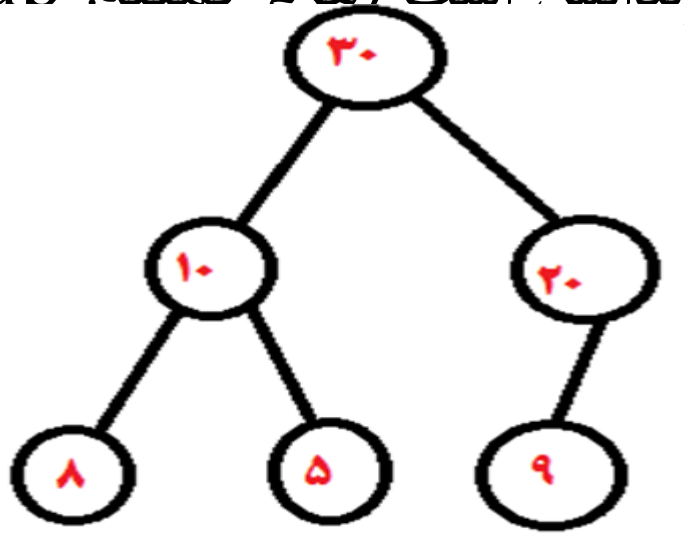
یادآوری مبحث هرم (Heap)

- هرم یک درخت دودویی با ویژگیهای زیر است:

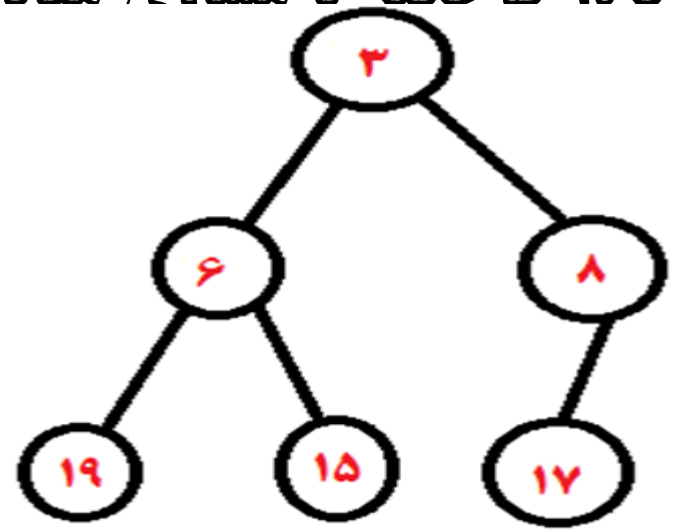
۱- کامل است.

۲-۱: مقدار هر گره، بزرگتر یا مساوی مقادیر گره های فرزندانش است (هرم بیشینه: MaxHeap).

۲-۲: مقدار هر گره، کوچکتر یا مساوی مقادیر گره های فرزندانش است (هرم کمینه: MinHeap).

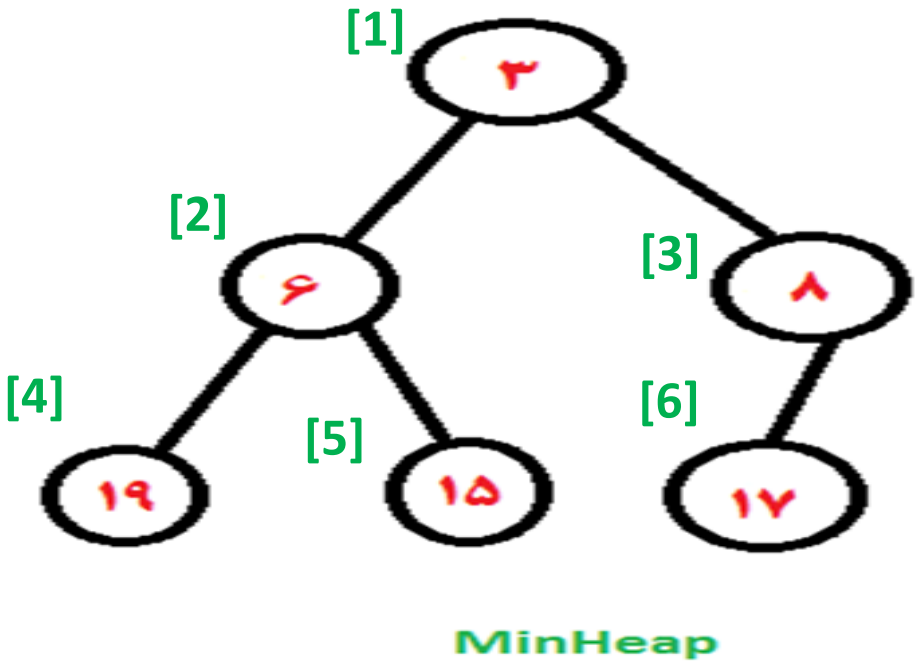


MaxHeap



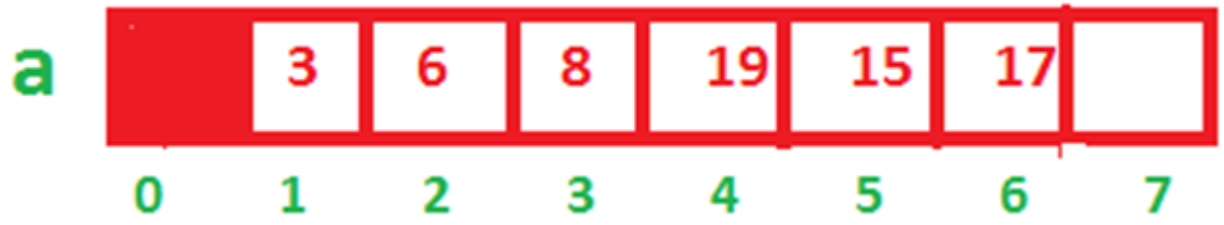
MinHeap

- هرم را با آرایه پیاده سازی می کنند.
  - ✓ a[0] خالی می ماند.
  - ✓ ریشه در a[1] قرار می گیرد.
  - ✓ برای هر گره در مکان k : فرزندان چپ و راست (در صورت وجود) به ترتیب در مکان های 2k و 2k+1 قرار می گیرند.



MinHeap

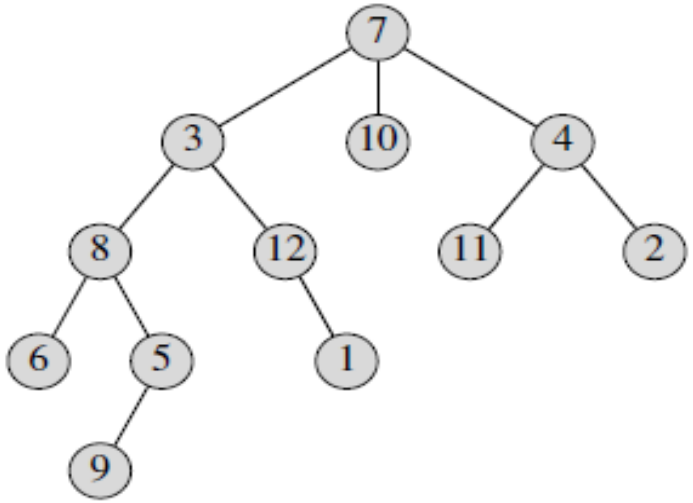
- بعنوان مثال :

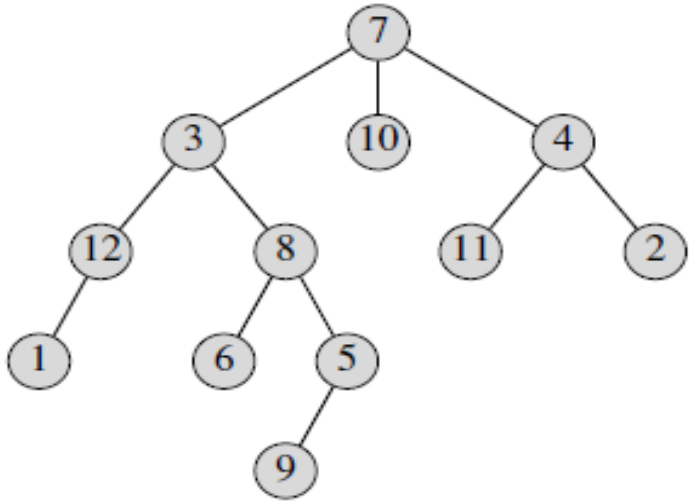


- یک هرم با n گره دارای ارتفاع $\theta(\log n)$ می باشد چون درخت کامل می باشد.
- پیدا کردن کوچکترین عنصر در یک MinHeap و یا بزرگترین عنصر در یک MaxHeap دارای پیچیدگی $\theta(1)$ خواهد بود.
- پیدا کردن بزرگترین عنصر در یک MinHeap و یا کوچکترین عنصر در یک MaxHeap دارای پیچیدگی $\theta(n)$ خواهد بود.
- اگر n تعداد گره ها باشد پیچیدگی تابع درج در بدترین حالت از مرتبه $O(\log n)$ و در بهترین حالت از مرتبه $\Omega(1)$ خواهد بود.
- اگر n تعداد گره ها باشد پیچیدگی تابع حذف در بدترین حالت از مرتبه $O(\log n)$ و در بهترین حالت از مرتبه $\Omega(1)$ خواهد بود.

یادآوری: درخت ریشه دار (Rooted Tree) و درخت مرتب شده (Ordered Trees)

- ✓ درخت ریشه دار: درختی که در آن، یکی از گره ها ریشه باشد.
- ✓ درخت مرتب شده: این نوع درخت، یک درخت ریشه داری است که در آن، فرزندان هر گره، مرتب شده هستند به این مفهوم که اگر گرهی، k فرزند داشته باشد آنگاه فرزندان بصورت اولین فرزند، دومین فرزند و .... نامگذاری می شوند.
- ✓ بعنوان مثال: اگر دو درخت زیر بعنوان درخت های مرتب شده در نظر گرفته شوند با هم متفاوت خواهند بود چون فرزندان گره ۳ ترتیب های متفاوتی دارند. در اولی، ۸ اولین فرزند است در حالیکه در دومی، دومین فرزند است.



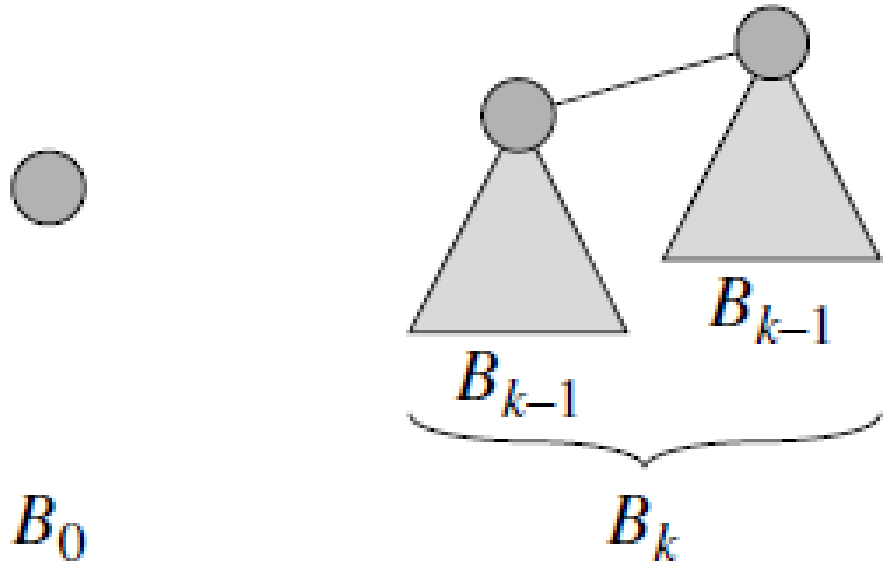


- ✓ نکته: اگر دو درخت مذکور، بعنوان درخت های غیر مرتب شده در نظر گرفته شوند با هم یکسان خواهند بود.

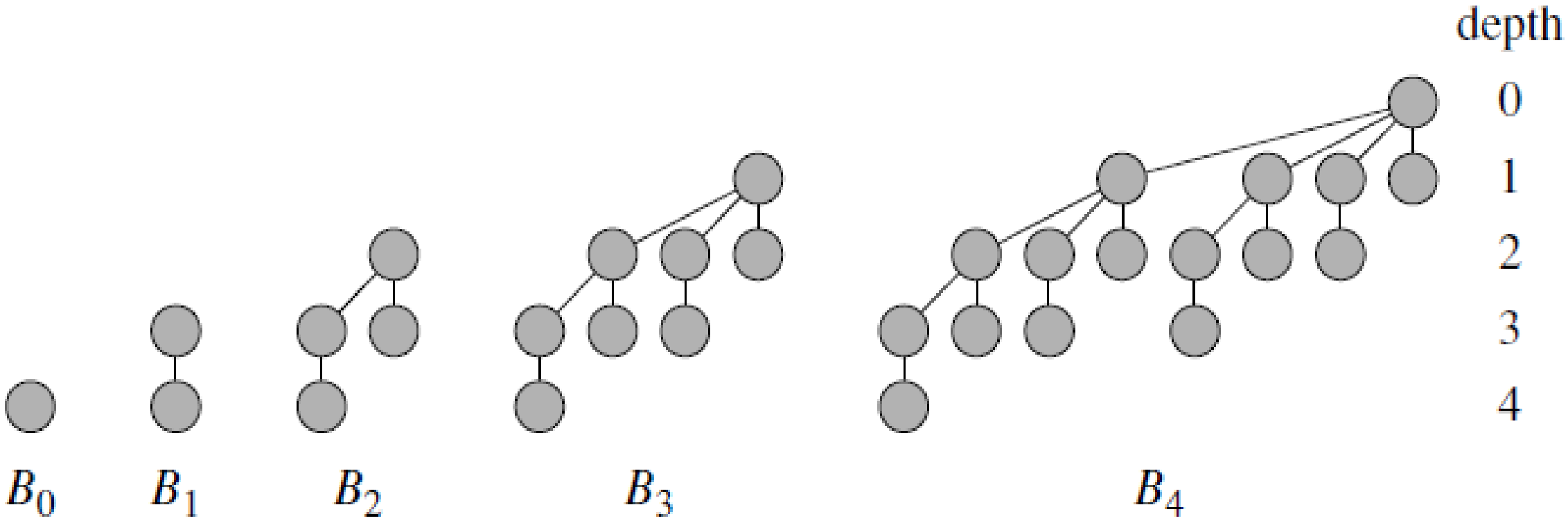
درخت های دوجمله ای (Binomial Trees)

✓ تعریف: درخت دوجمله ای B_k یک درخت مرتب شده است که بطور بازگشتی و بصورت زیر تعریف می شود:

- درخت دوجمله ای B_0 از یک گره تشکیل شده است.
- درخت دوجمله ای B_k از دو درخت دوجمله ای B_{k-1} تشکیل شده است که ریشه یکی بعنوان سمت چپ ترین فرزند ریشه دیگر



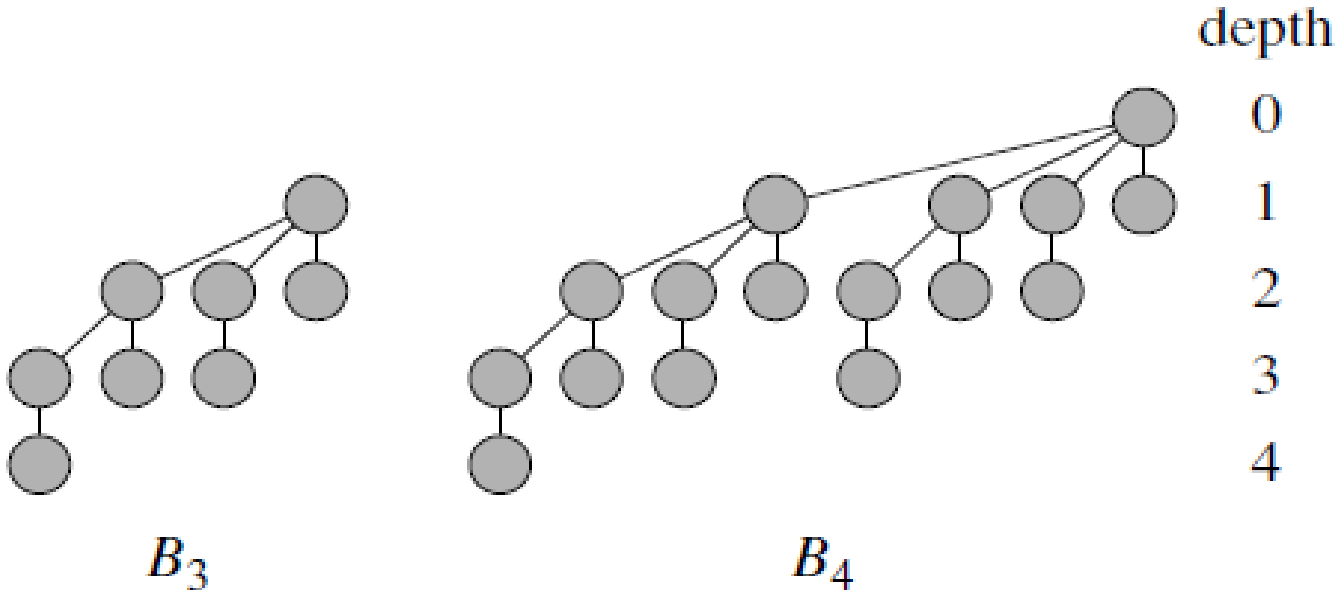
✓ مثال: شکل زیر درخت های دوجمله ای B_0 تا B_4 را نشان می دهد.



✓ **ویژگیهای درخت دوجمله ای: برای درخت دوجمله ای B_k داریم:**

۱- ارتفاع درخت برابر k است.

بعنوان مثال: ارتفاع درخت B_4 برابر ۴ است و ارتفاع درخت B_3 برابر ۳ است.



۲- در عمق i، دقیقاً $\binom{k}{i}$ گره وجود دارد.

بعنوان مثال: در درخت دوجمله ای B_4، در عمق 2، دقیقاً $\binom{4}{2} = 6$ گره وجود دارد.

نکته: این درخت، درخت دوجمله ای نامیده می شود زیرا ضرایب بسط $(x+y)^k$ است.

۳- 2^k گره وجود دارد که 2^{k-1} از آنها برگ هستند و بقیه گره های داخلی هستند.

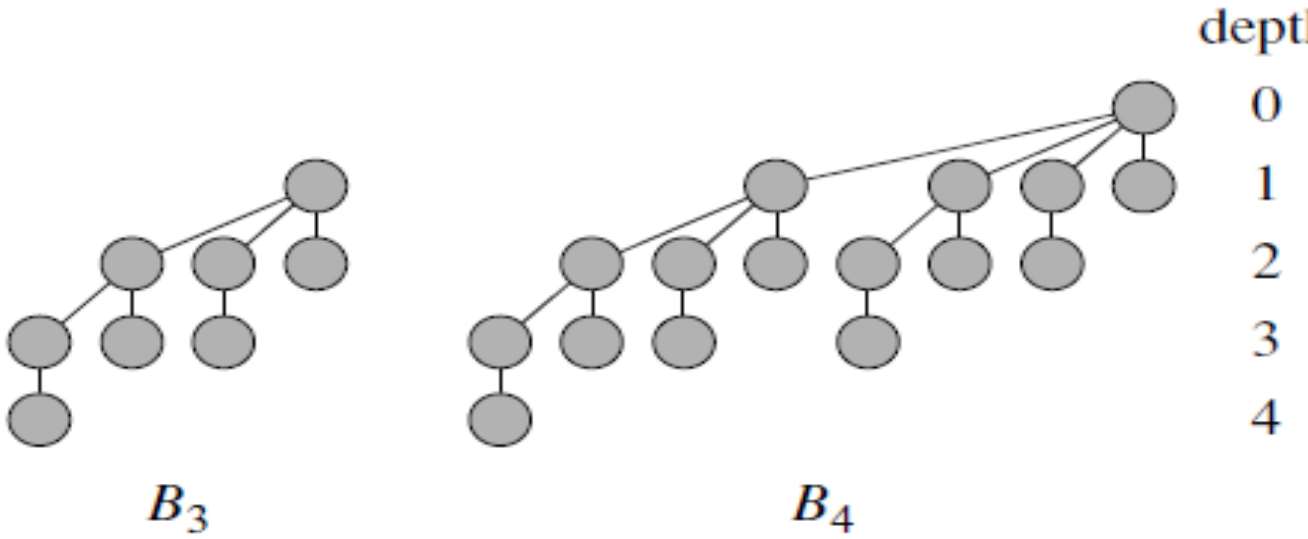
درستی این موارد با استقراء انجام می شوند: برای تعداد گره ها، چون درخت دوجمله ای B_k از دو درخت دوجمله ای B_{k-1} تشکیل شده است پس تعداد گره های درخت دوجمله ای B_k عبارت است از: $2^{k-1}+2^{k-1}=2^k$

بعنوان مثال: تعداد گره های درخت B_4 برابر $2^4=16$ است.

$$n=\binom{4}{0}+\binom{4}{1}+\binom{4}{2}+\binom{4}{3}+\binom{4}{4}=1+4+6+4+1=16$$

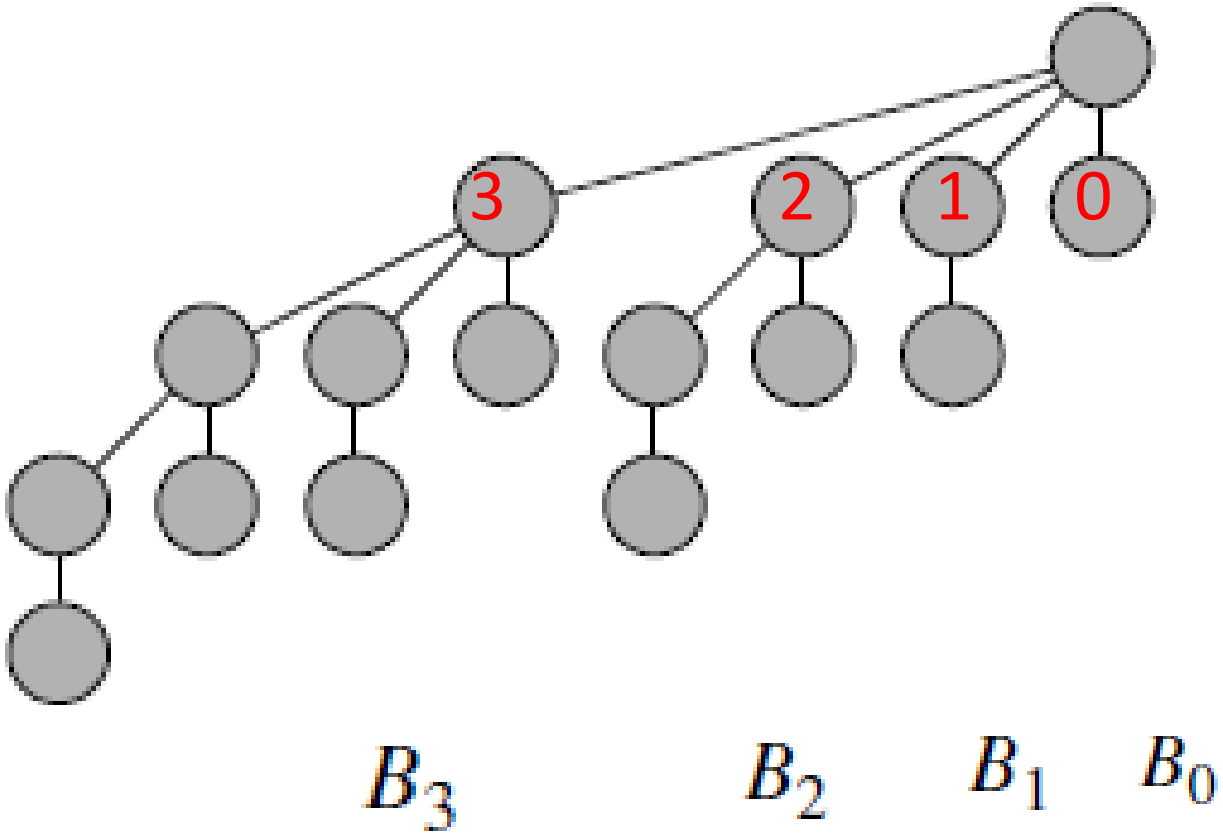
۴- ریشه دارای درجه k است که از درجه گره های دیگر بزرگتر است.

بعنوان مثال: درجه رخت B_3 برابر ۳ است.



۵- اگر فرزندان ریشه از راست به چپ با 0 ، 1، ...، k-1 شماره گذاری شوند آنگاه i ریشه زیردرخت دوجمله ای B_i خواهد بود.

بعنوان مثال: درخت دوجمله ای B_4 را در نظر بگیرید.



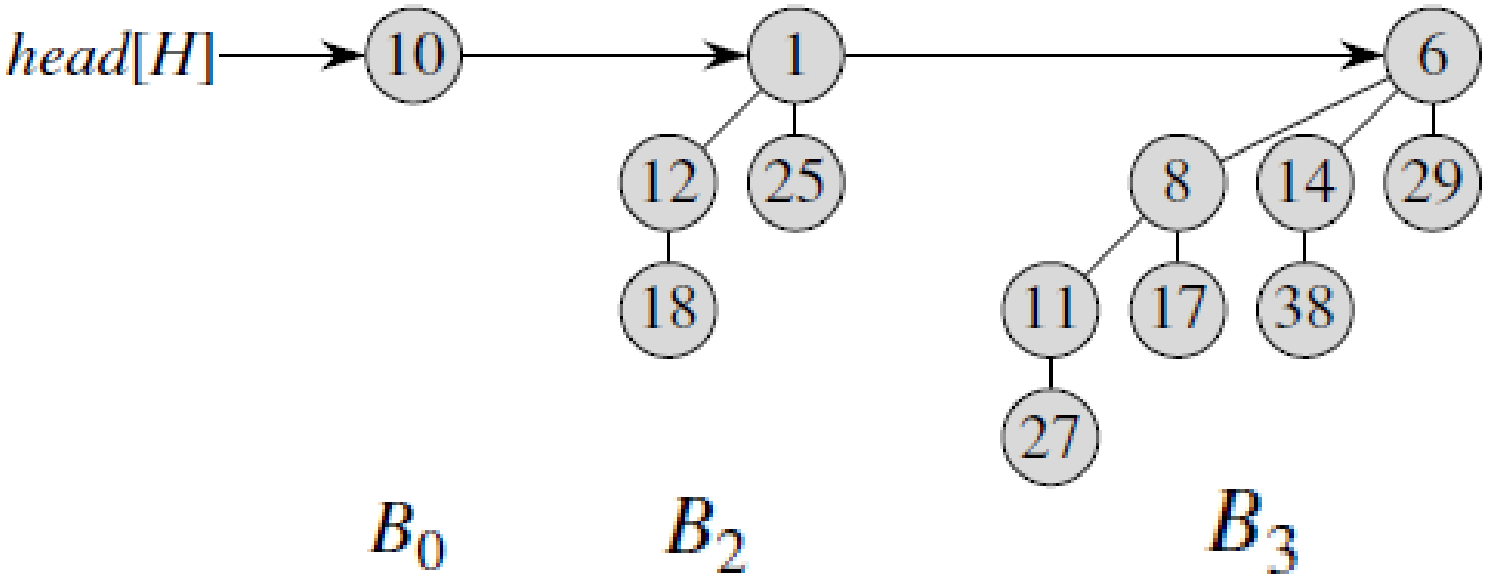
هرم های دوجمله ای (Binomial Heap)

✓ تعریف: هرم دوجمله ای H، مجموعه ای از درخت های دوجمله ای است که دارای ویژگیهای زیر می باشد:

۱- هر درخت دوجمله ای در H از ویژگی MinHeap پیروی می کند. یعنی، کلید یک گره کوچکتر یا مساوی کلید فرزندانش می باشد.

۲- برای عدد صحیح غیرمنفی k، حداکثر یک درخت دوجمله ای با درجه k در H وجود دارد.

بعنوان مثال: شکل زیر

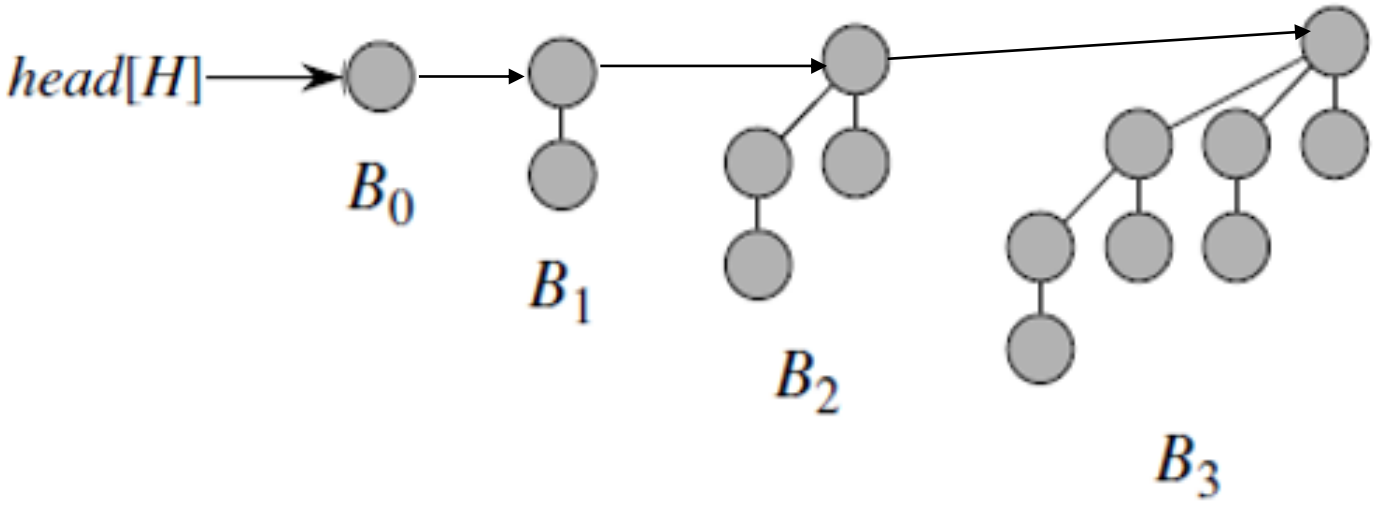


سوال: ثابت کنید که هرم دوجمله ای با n گره شامل حداکثر $\lfloor logn \rfloor + 1$ درخت دوجمله ای است.

اثبات: فرضاً تعداد درختان دو جمله ای برابر p باشد. چون هر درخت دوجمله ای B_k دارای 2^k گره است بنابراین در بدترین حالت، همه درخت های دو جمله ای از B_0 تا B_{p-1} باید در درخت موجود باشند. بنابراین:

$$2^0 + 2^1 + \cdots + 2^{p-1} = 2^p - 1 = n \rightarrow 2^p = n+1 \rightarrow p = \lfloor logn \rfloor + 1$$

بعنوان مثال: هرم دوجمله

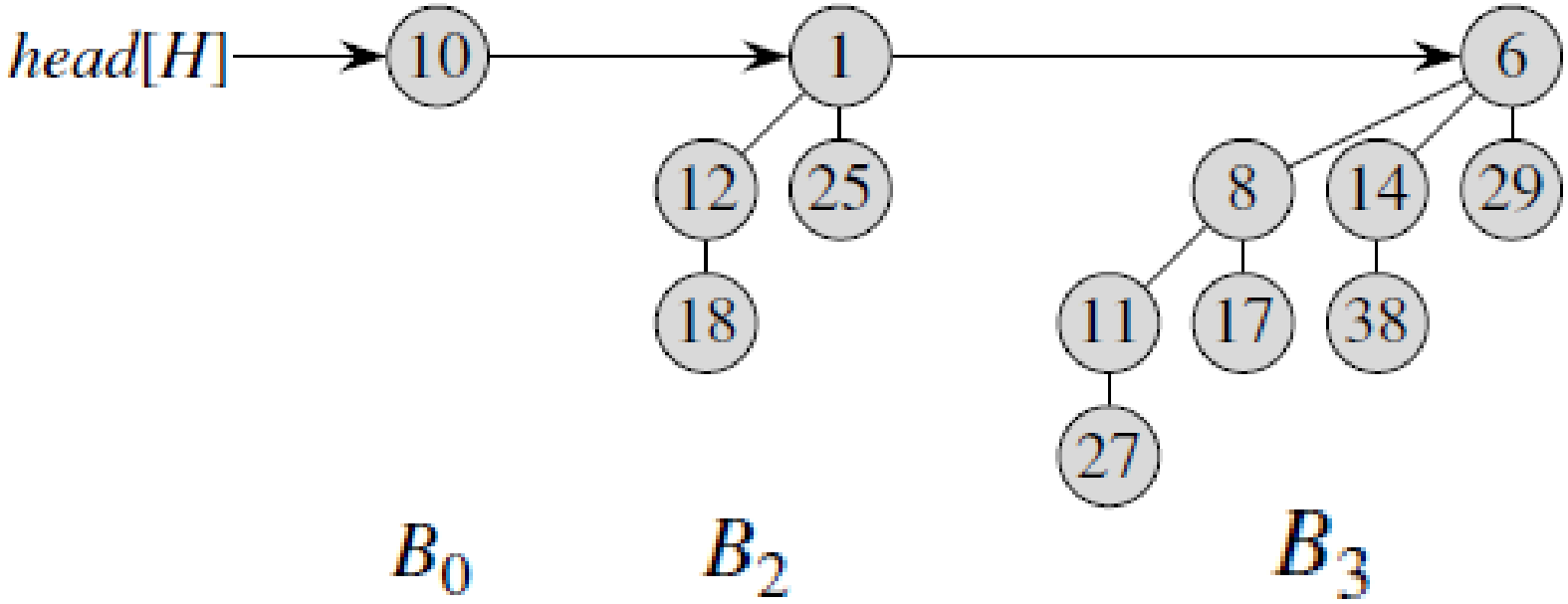


$$+ 1 = 3 + 1 = 4$$

کار در کلاس: یک هرم دوجمله ای با ۱۳ گره رسم کرده و آن را با اعداد دلخواه پر کنید.

جواب:

$13=(1101)_2 \rightarrow 13=2^3+2^2+2^0 \rightarrow B_0, B_2, B_3$



نمایش (ذخیره) هرم های دوجمله ای: هر درخت دوجمله ای در هرم، با نمایش فرزند چپ (left-child)، همزاد راست (right-sibling) نمایش داده می شود.

در واقع، هر گره در هرم دوجمله ای بصورت زیر تعریف می شود:

اشاره گر به والد

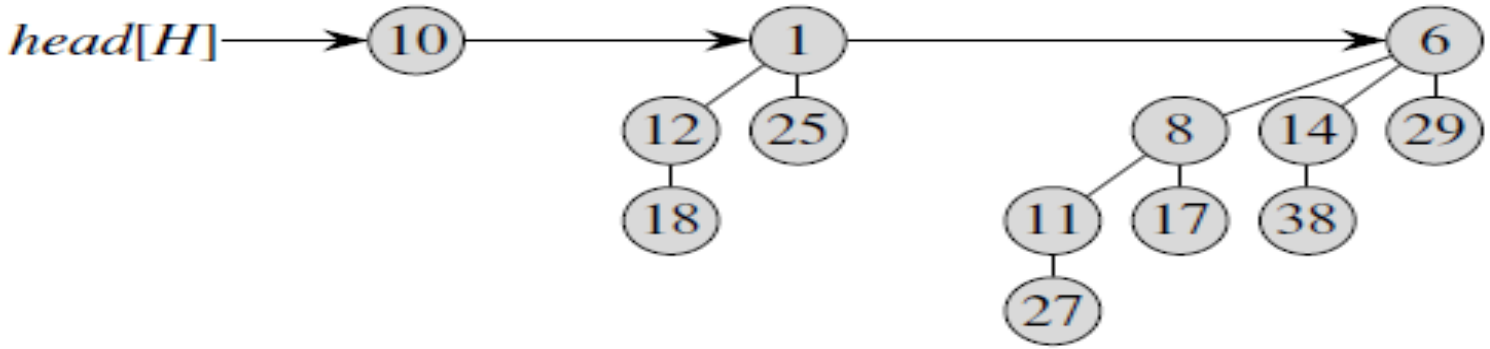
کلید

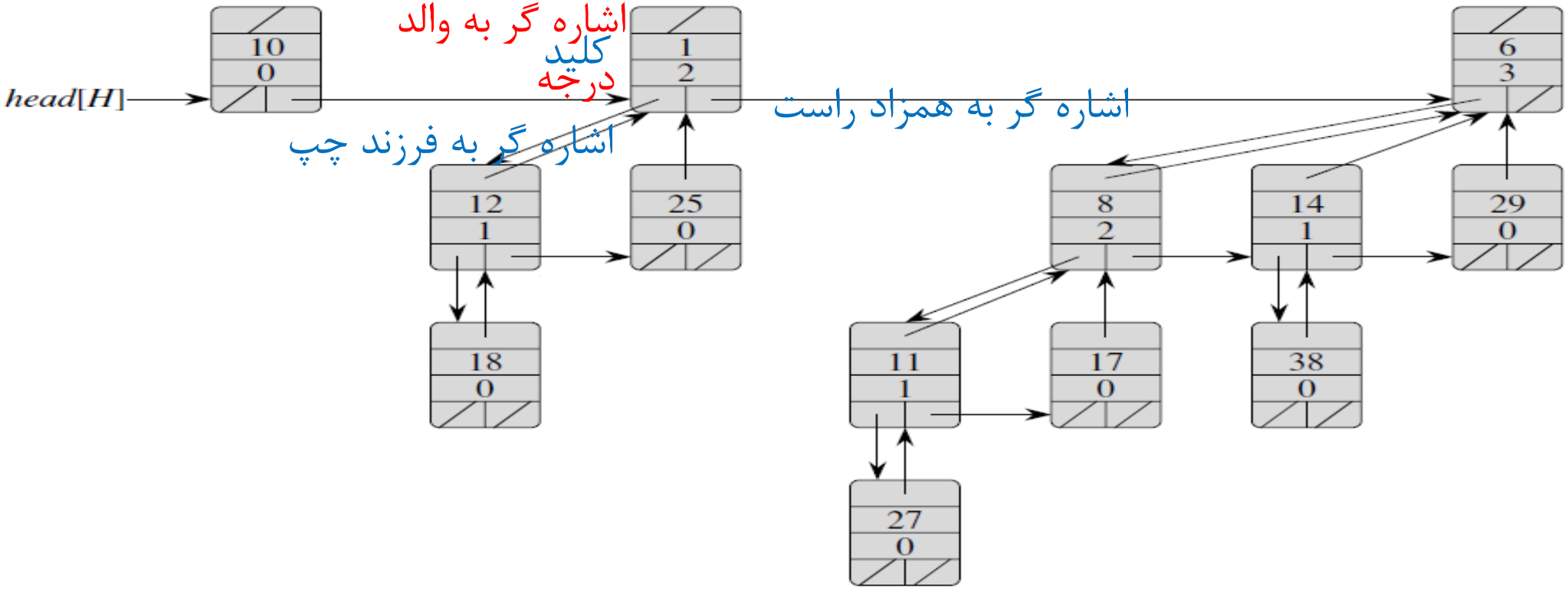
درجه

اشاره گر به چپترین فرزند

اشاره گر به همزاد راست

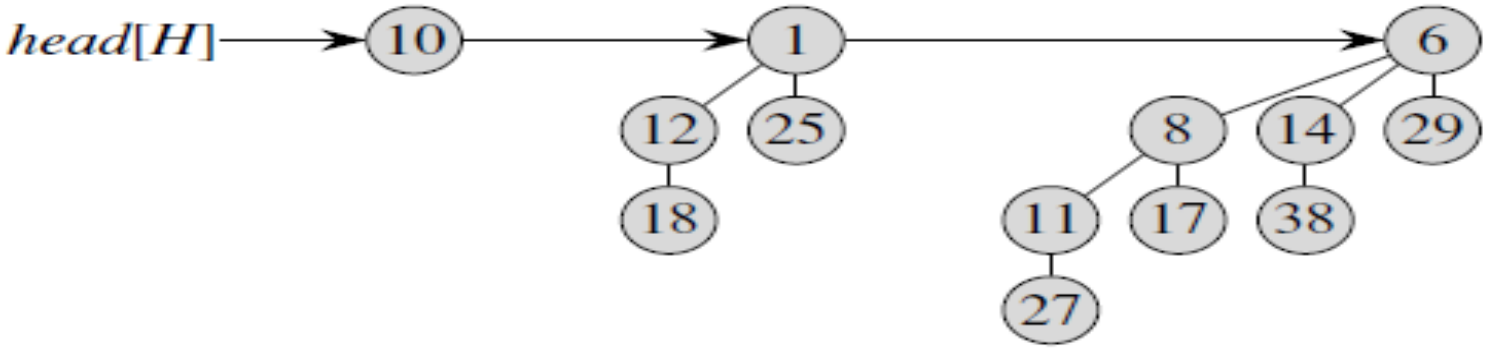
بعنوان مثال: نمایش هرم دو جمله ای زیر را در ادامه می بینید:





کار در کلاس: عناصر هرم دو جمله ای زیر را به ترتیب VLS چاپ کنید.

همزاد راست ، فرزند چپ، خود گره



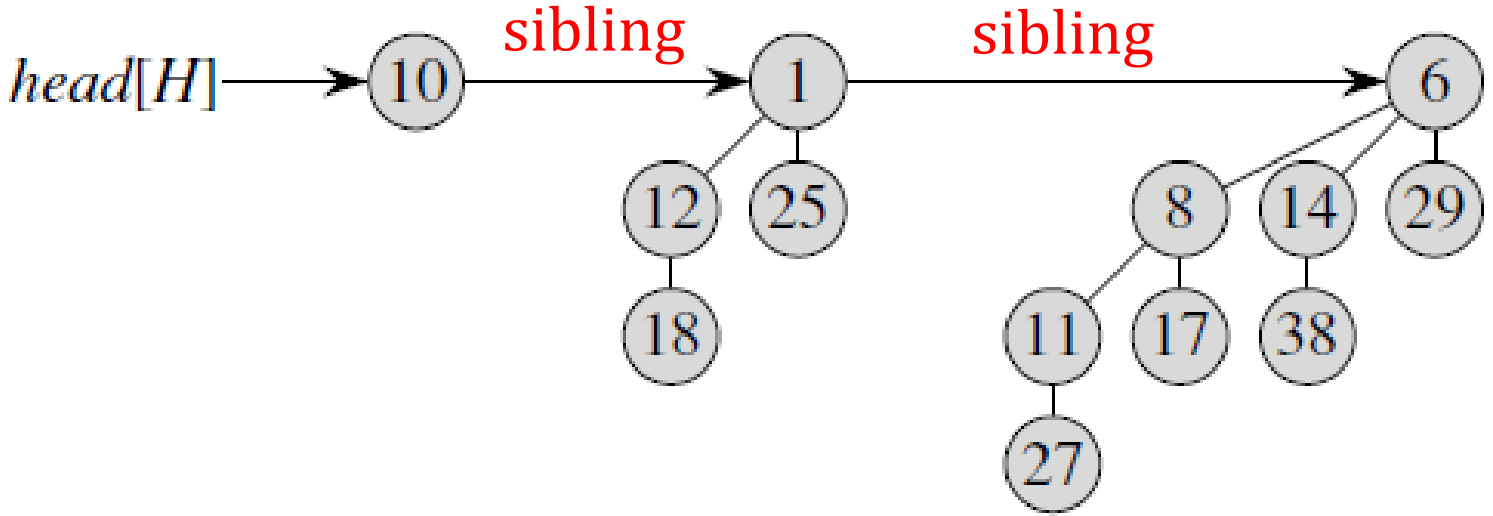
جواب:

10, 1, 12, 18, 25, 6, 8, 11, 27, 17, 14, 38, 29

۱- پیدا کردن کوچکترین کلید

چون کوچکترین کلیدها در ریشه درختان دوجمله ای هستند و تعداد آنها در بدترین حالت برابر $\lfloor logn \rfloor + 1$ است بنابراین می توان در بدترین حالت با زمان $O(logn)$ کوچکترین عنصر را پیدا کرد.

بعنوان مثال: در هرم زیر

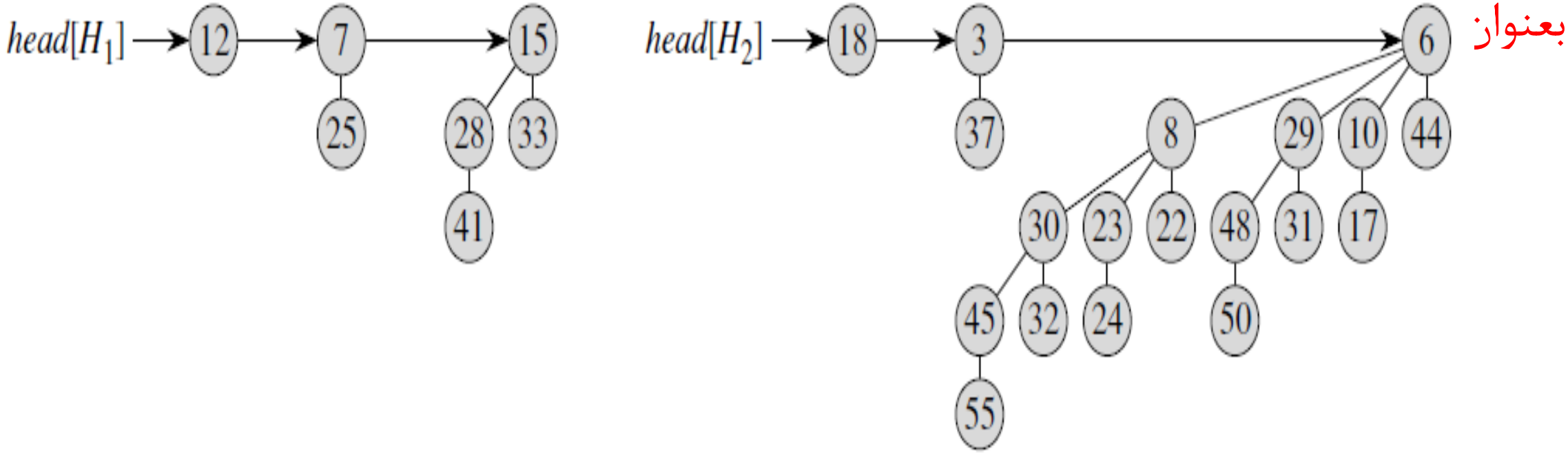


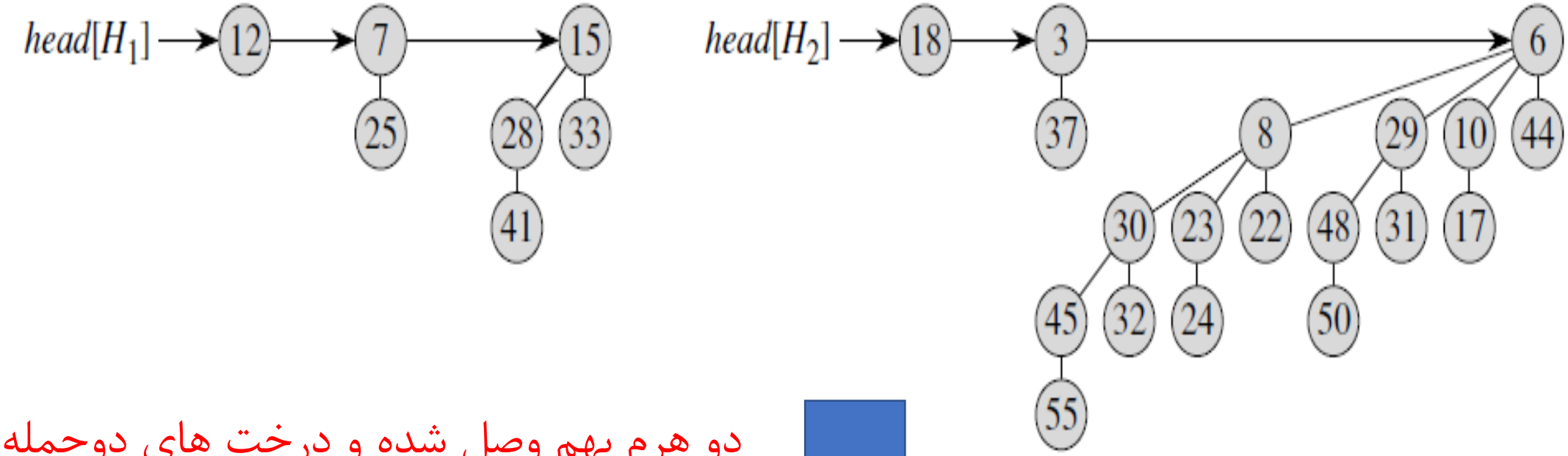
و 6 می باشد.

۲- واحد سازی (اجتماع: Union) دو هرم دوجمله ای

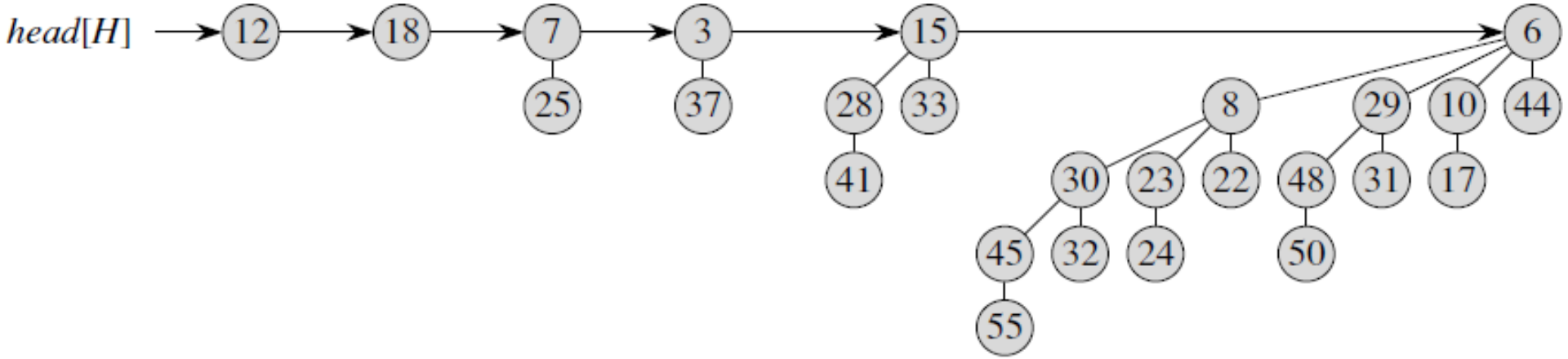
این عمل، درخت های دو جمله ای را که ریشه هایشان دارای درجه یکسانی می باشند را به هم متصل کرده و این عمل را تا موقعی انجام می دهد که دو درخت دوجمله ای با درجه یکسان موجود نباشند.

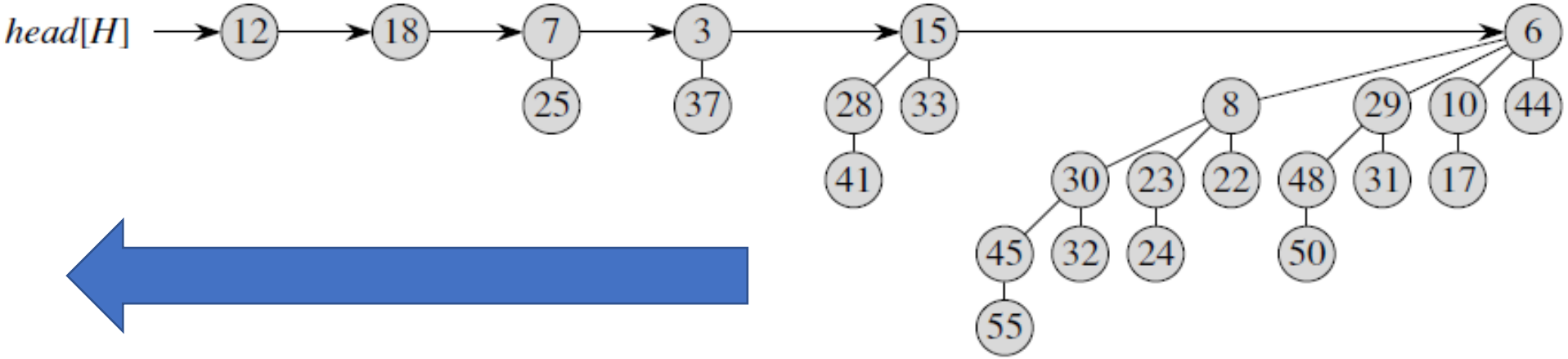
بعنوان



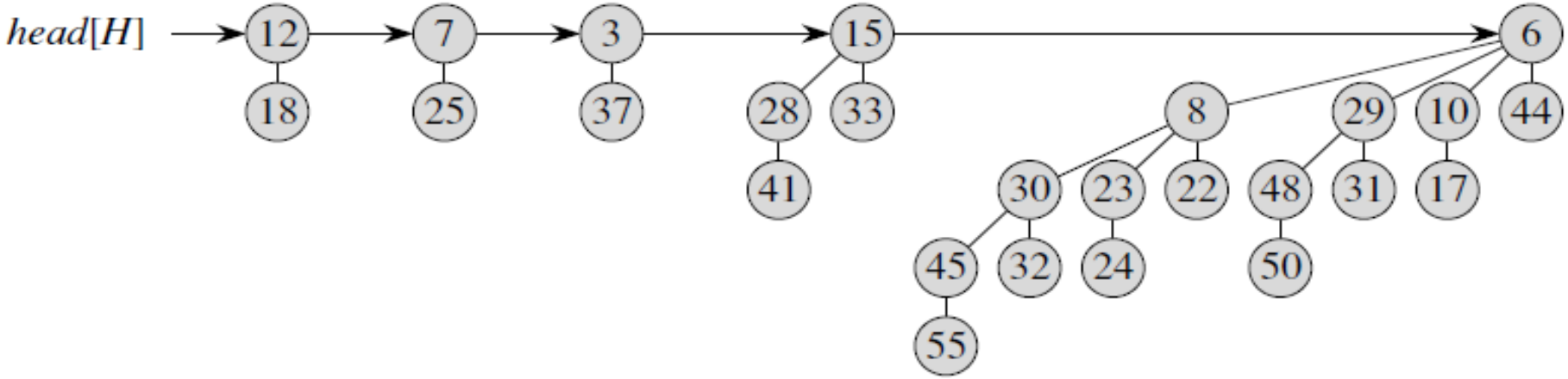


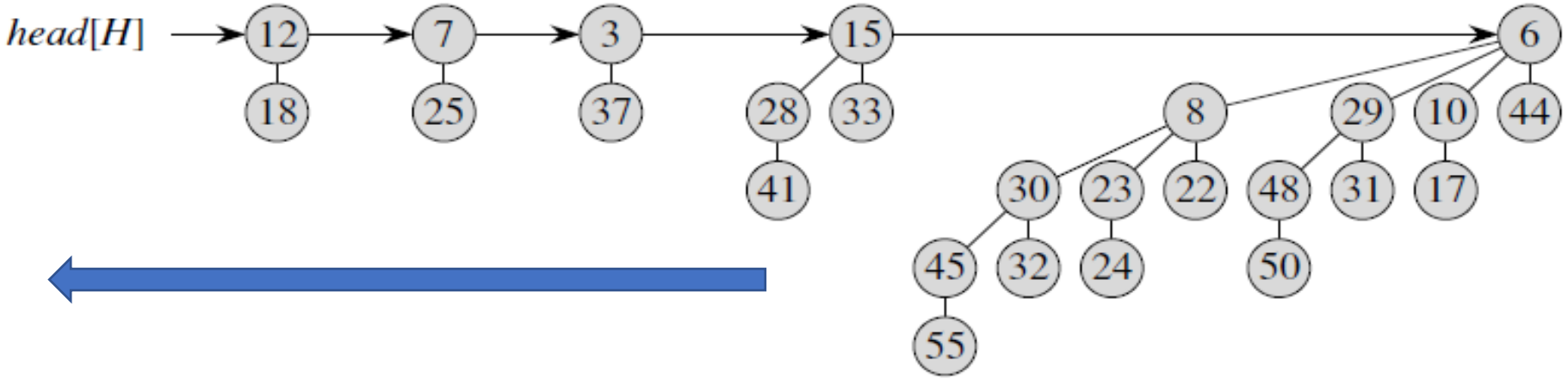
دو هرم بهم وصل شده و درخت های دوجمله ای از درجه پایین به بالا مرتب می شوند.



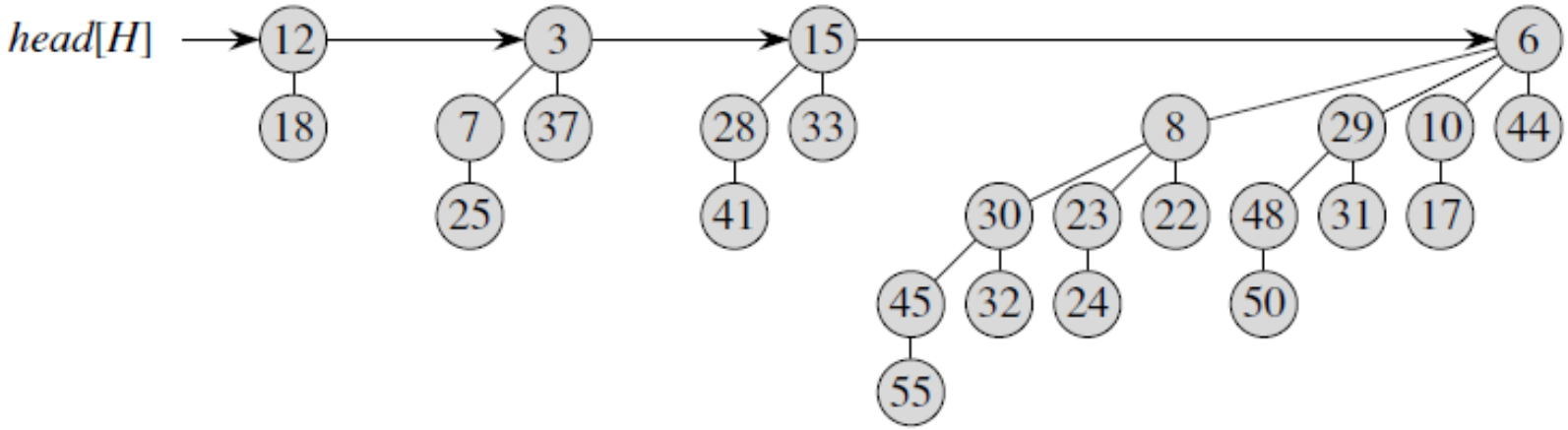


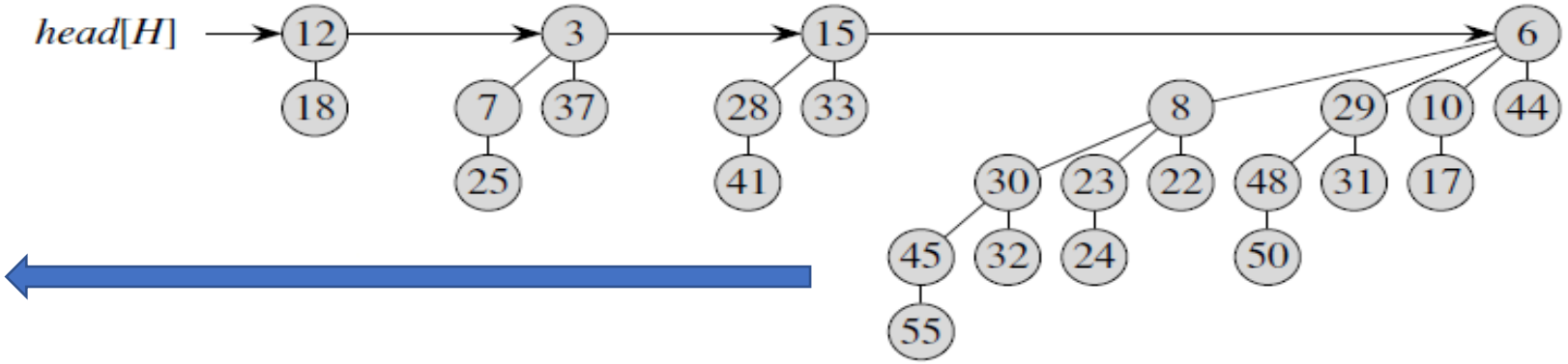
تمام درخت های دوجمله ای از درجه صفر باهم ادغام می شوند. ادغام از آخر هرم به ابتدای هرم انجام می شود.



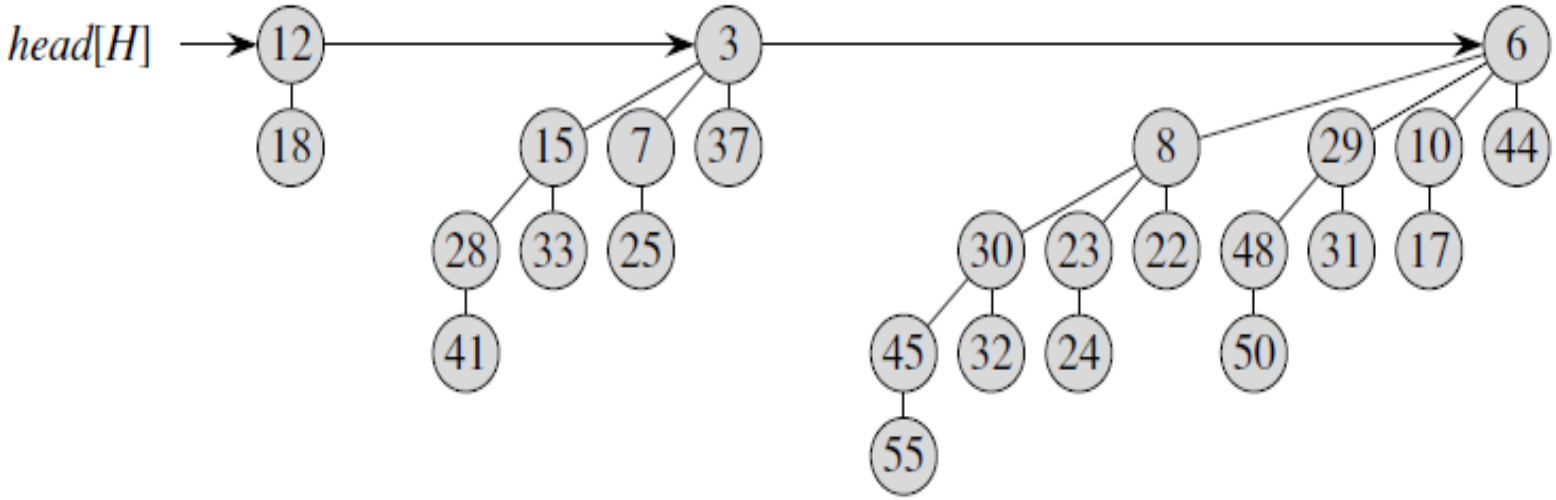


تمام درخت های دوجمله ای از درجه یک باهم ادغام می شوند. . ادغام از آخر هرم به ابتدای هرم انجام می شود.





تمام درخت های دوجمله ای از درجه دو باهم ادغام می شوند. . ادغام از آخر هرم به ابتدای هرم انجام می شود.

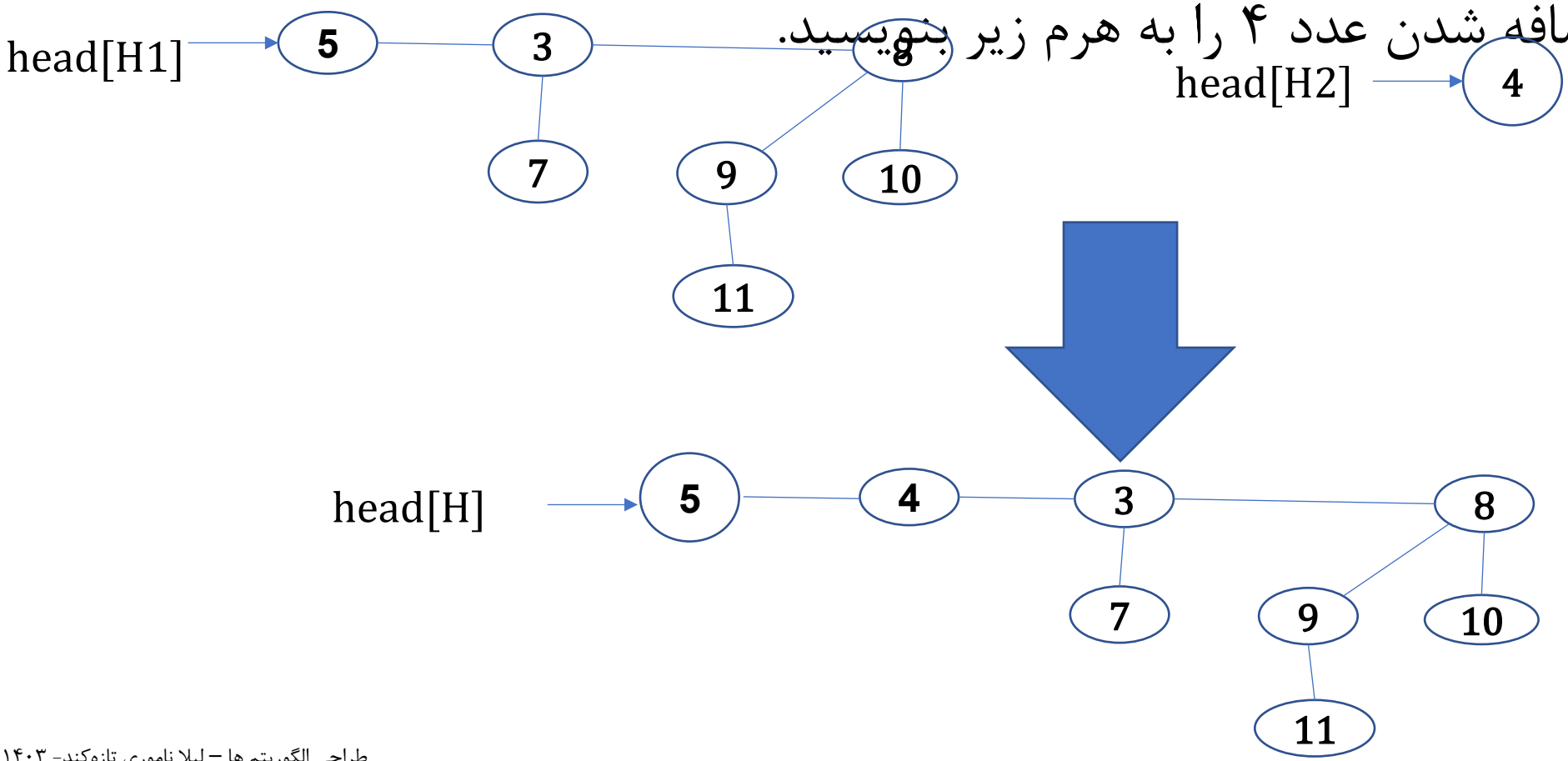


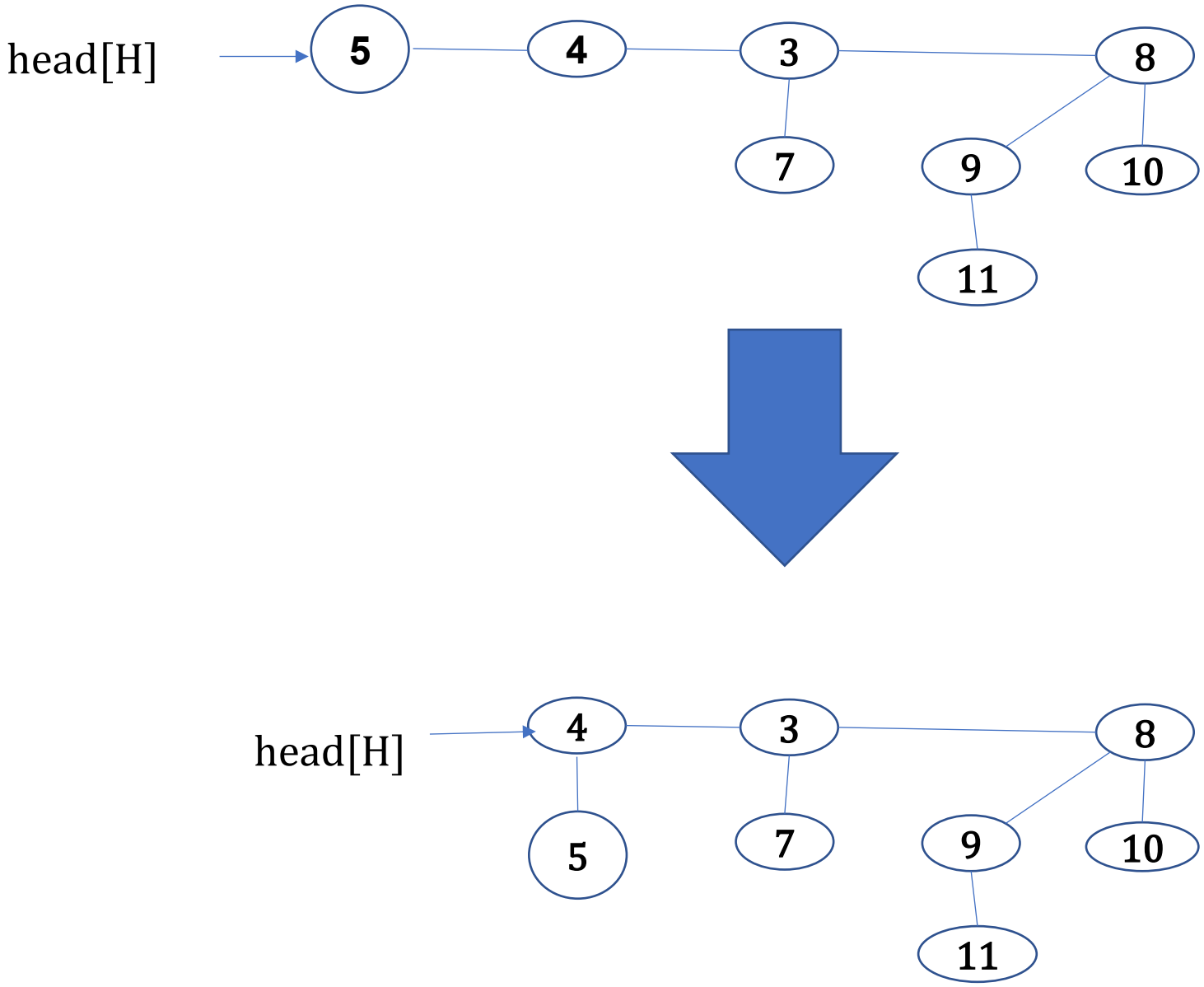
نکته: زمان اجرای عمل اجتماع دو هرم برابر است با $O(logn)$

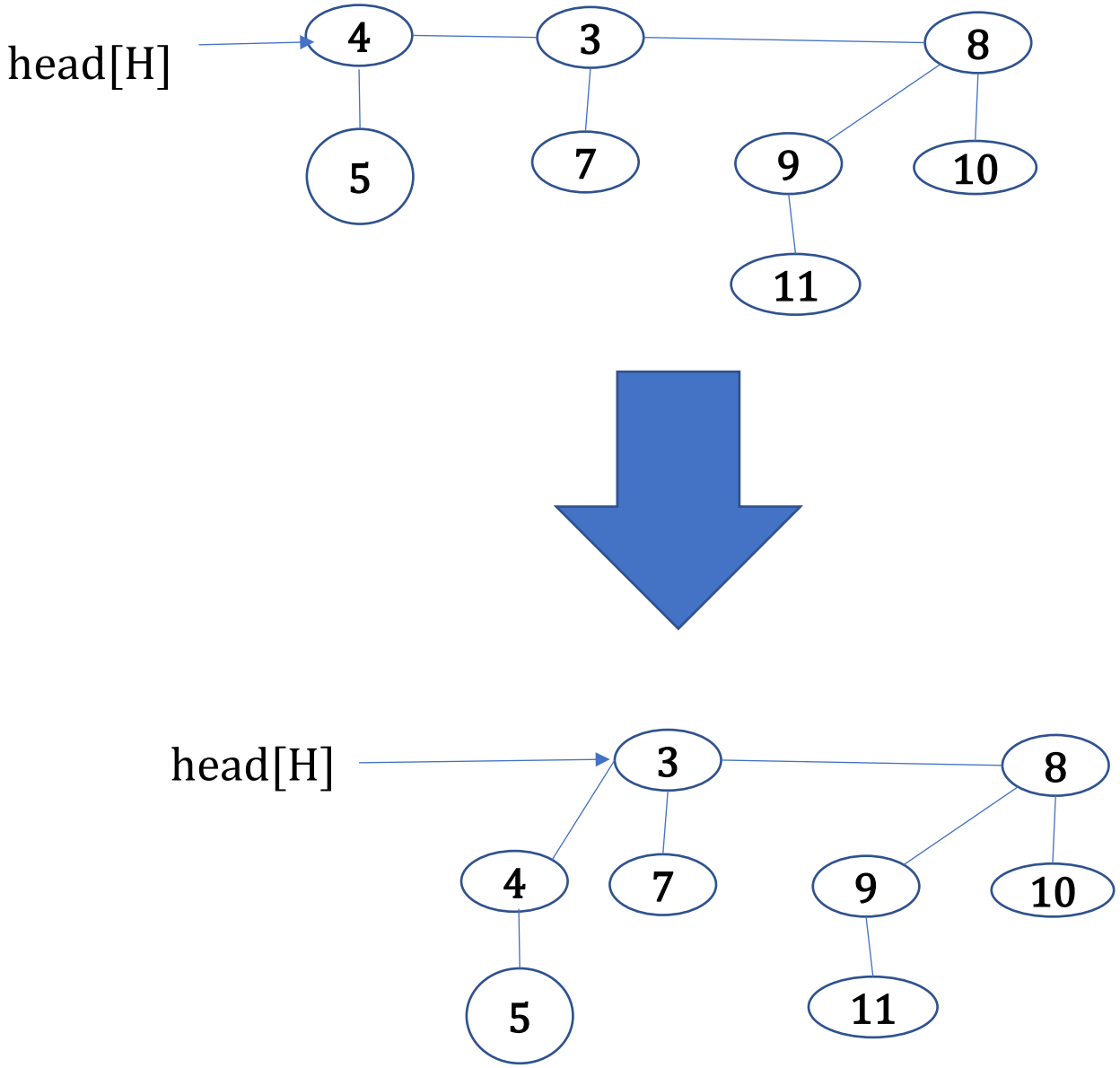
۳- درج عنصر جدید x به هرم دوجمله ای H1

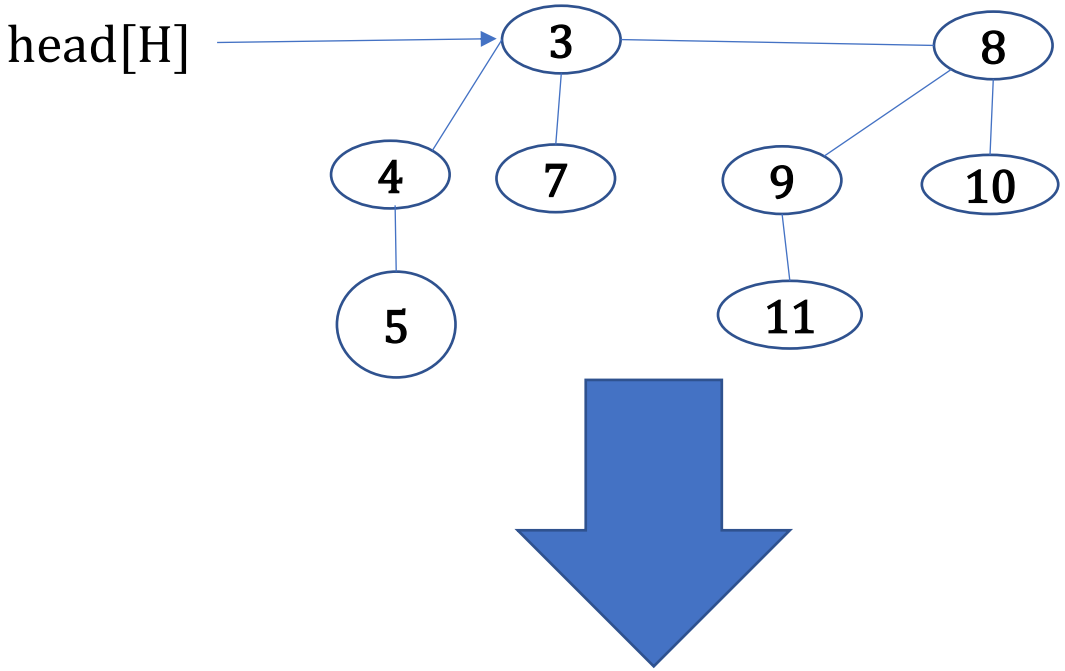
ابتدا با عنصر x یک هرم تک عنصری H2 ساخته سپس آنها را با عمل اجتماع به یک هرم واحد H تبدیل می کنیم.

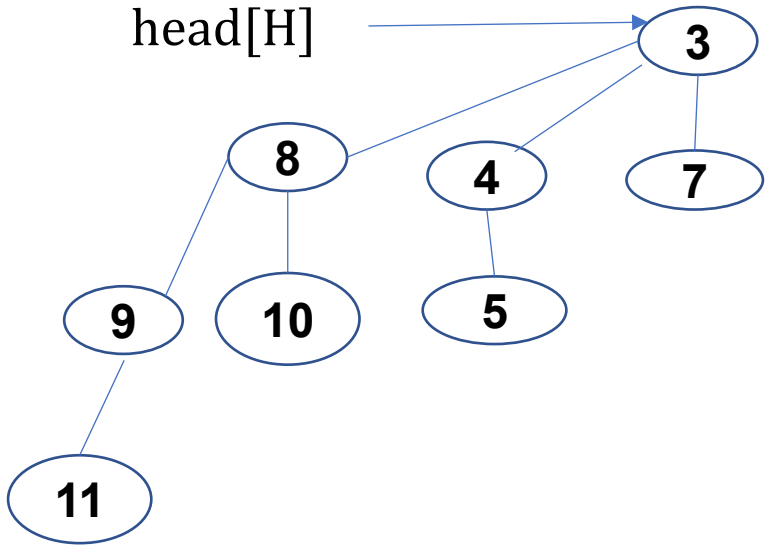
بعنوان مثال: مراحل اضافه شدن عدد ۴ را به هرم زیر بنویسید.




head[H]
5
4
3
8
7
9
10
11
head[H]
4
3
8
5
7
9
10
11


head[H]
4
3
8
5
7
9
10
11
head[H]
3
8
4
7
9
10
5
11

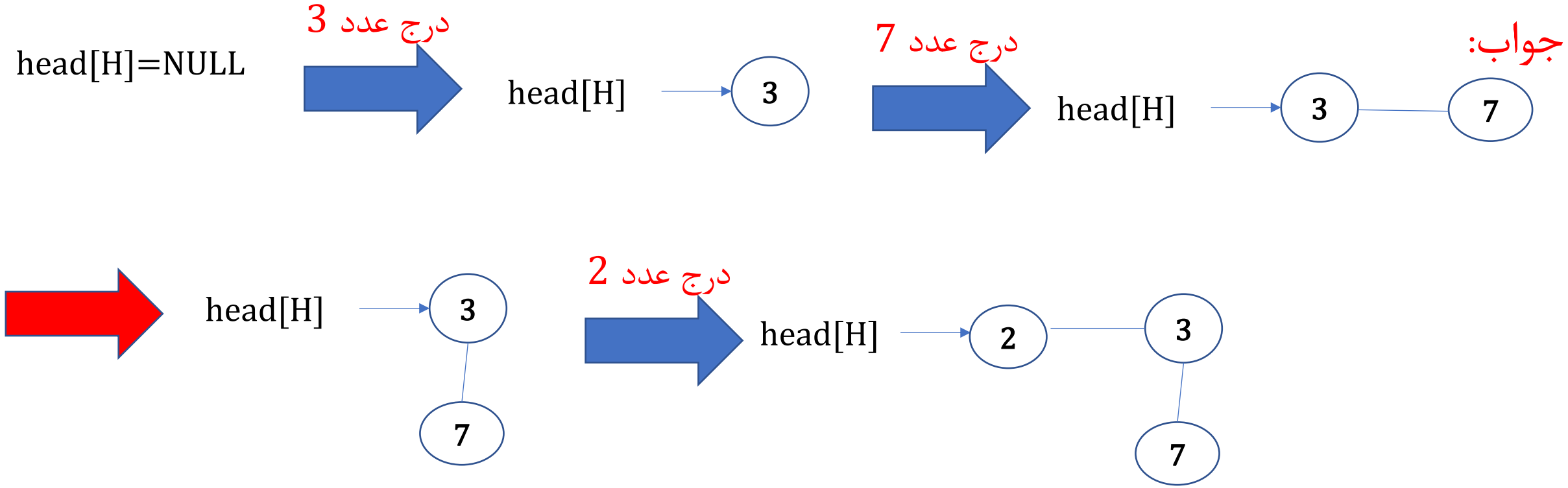


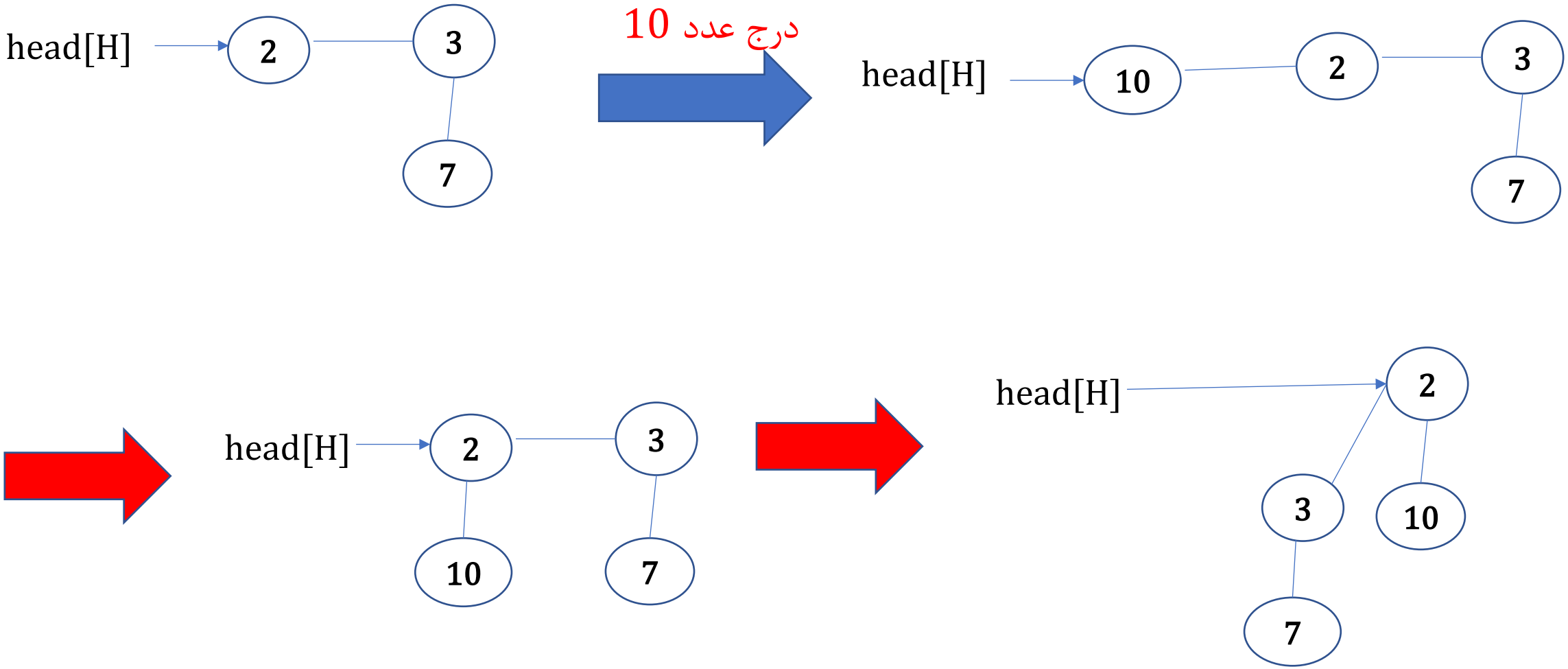


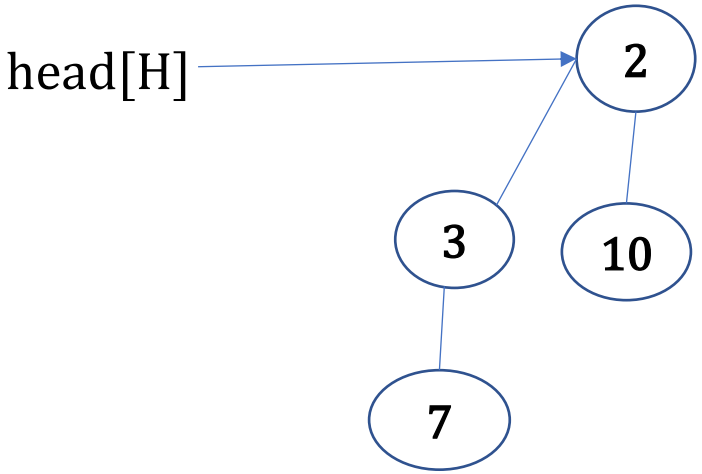
نکته: زمان اجرای عمل درج به هرم دوجله ای برابر است با $O(logn)$

کار در کلاس: با عناصر زیر یک هرم دوجمله ای بسازید.

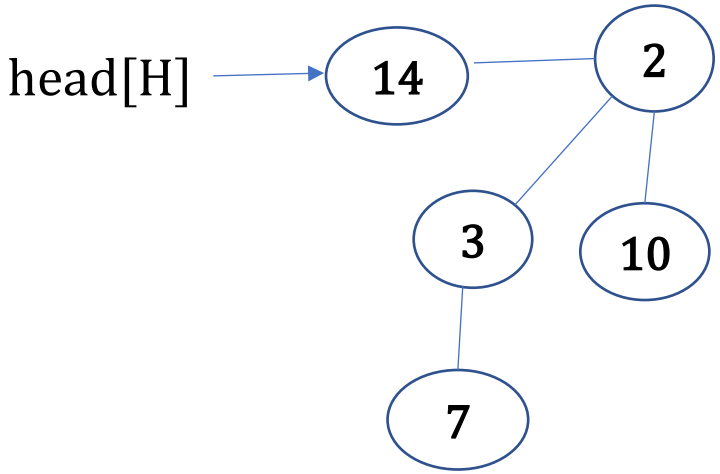
3, 7, 2, 10, 14, 5, 6



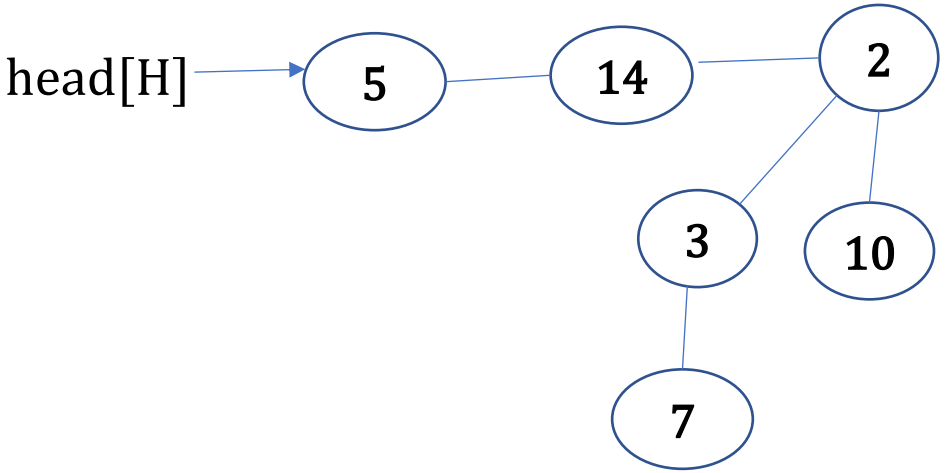

head[H]
2
3
7
درج عدد 10
head[H]
10
2
3
7
head[H]
2
3
10
7
head[H]
2
3
10
7

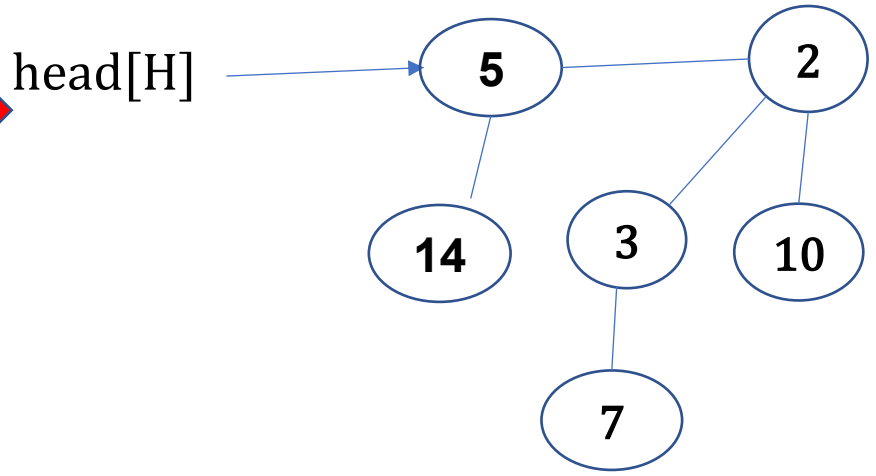


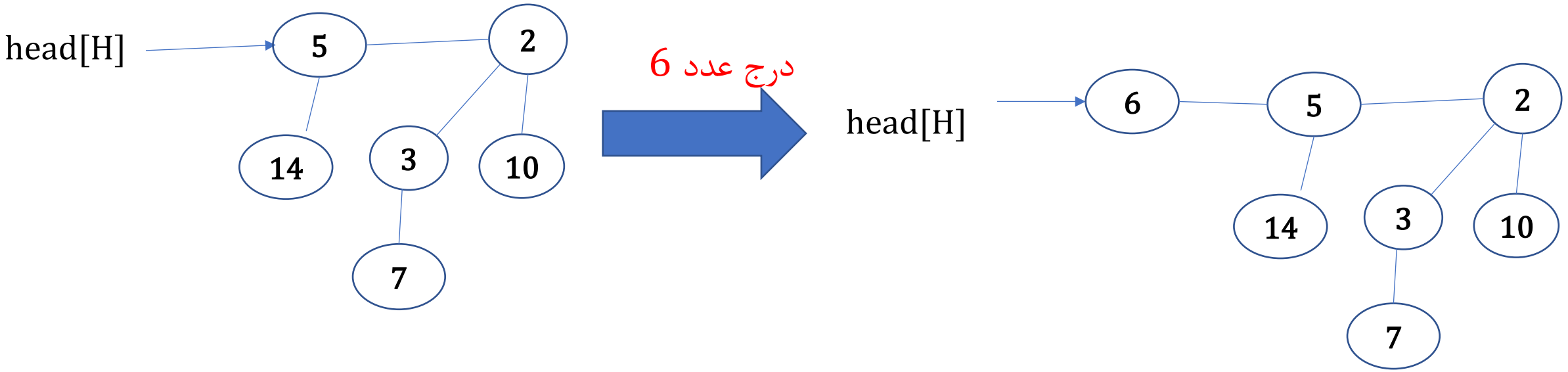
درج عدد 14



درج عدد 5



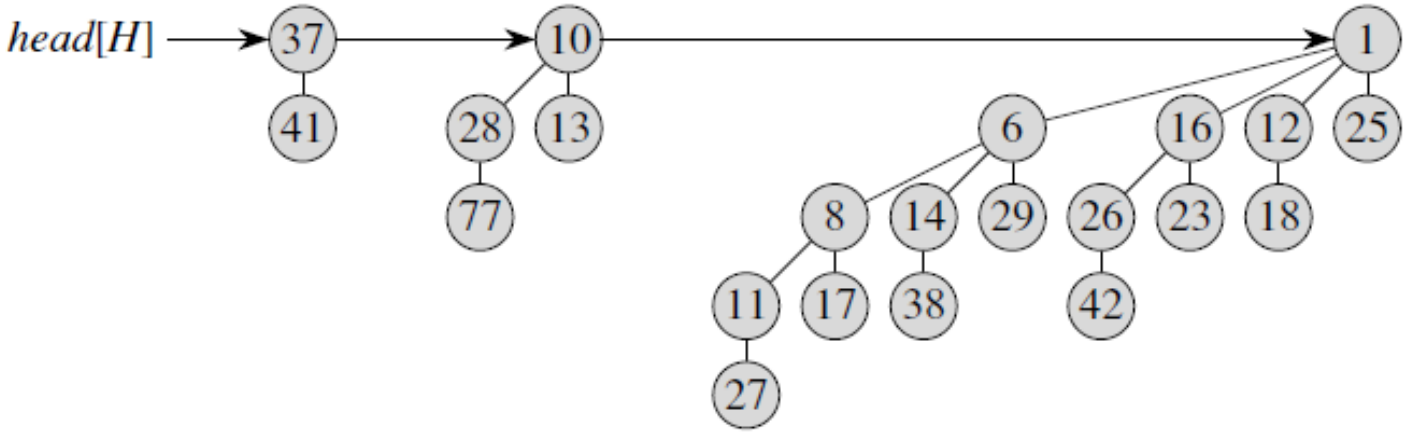



head[H]
5
2
14
3
10
7
درج عدد 6
head[H]
6
5
2
14
3
10
7

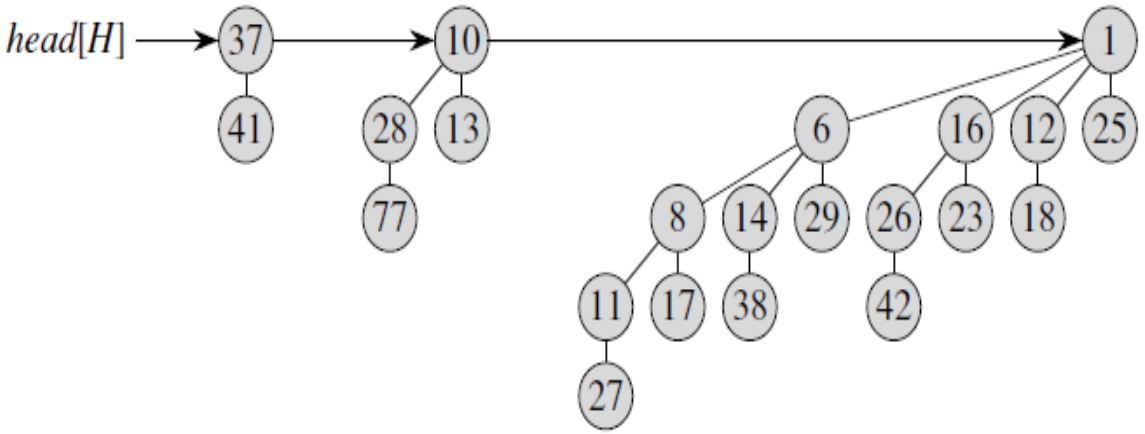
۴- استخراج گره با کمترین مقدار

این عمل، گره با کوچکترین مقدار را پیدا کرده و آن را از هرم حذف می کند.

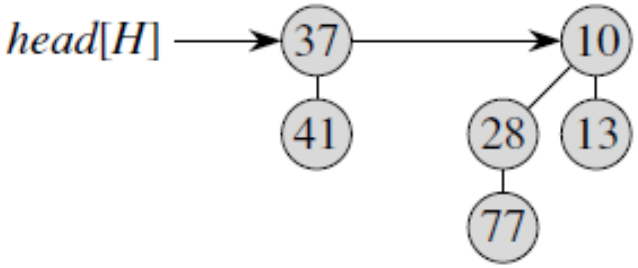
بعنوان مثال: مراحل استخراج گره با کمترین مقدار از هرم زیر

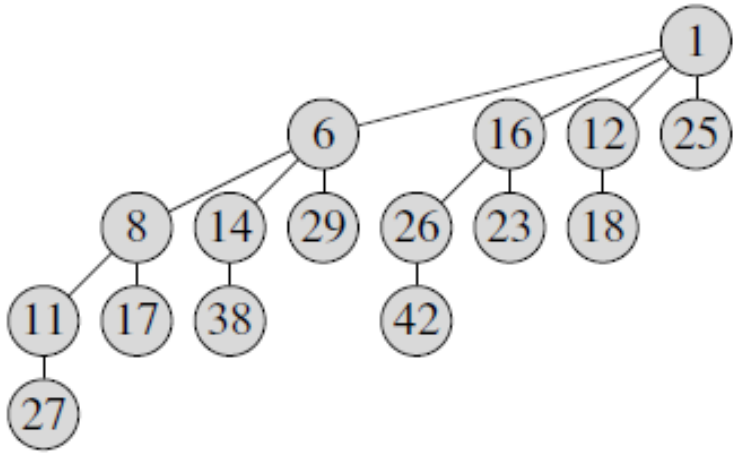


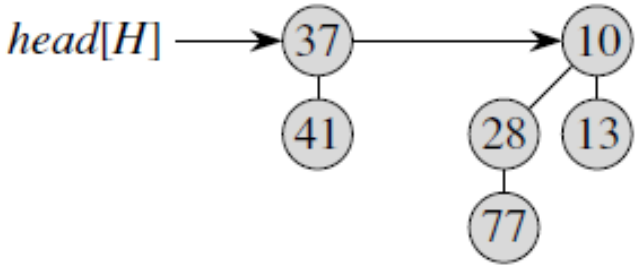
جواب: کوچکترین کلیدها در ریشه درختان دوجمله ای هستند که در اینجا گره با عدد ۱ می باشد.

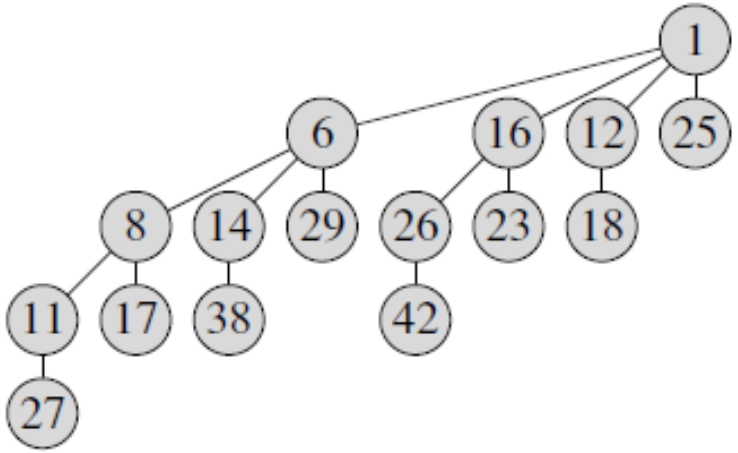


درخت دوجمله ای
مربوط به گره ۱ را از
هرم جدا می کنیم.

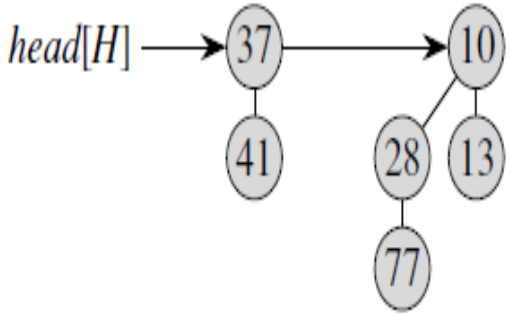


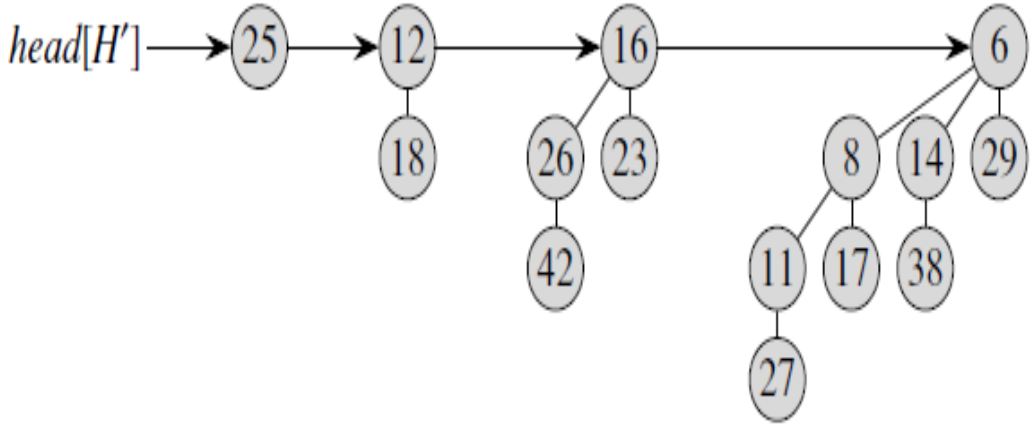


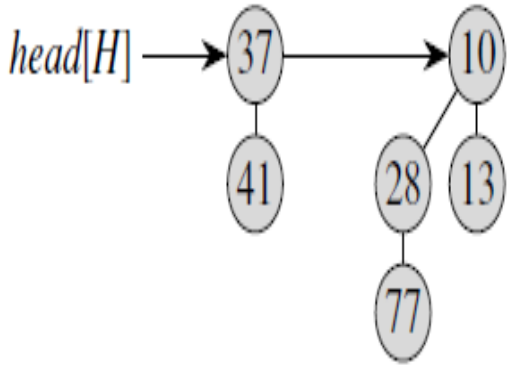


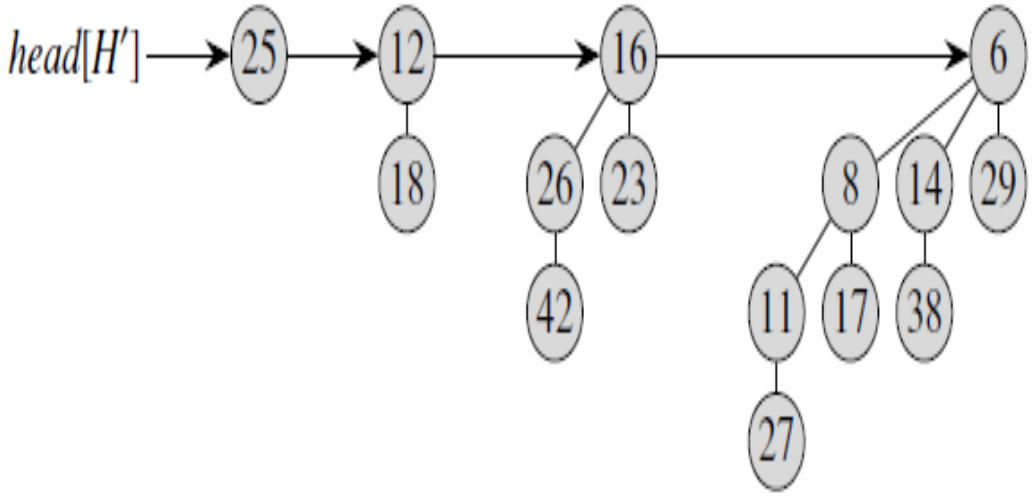


گره مربوط به عدد ۱ را حذف کرده و با فرزندان آن یک هرم جدید می سازیم. معمولا چون درختان دوجمله ای از درجه های کمتر شروع می شوند بنابراین ترتیب فرزندان را معکوس می کنیم.



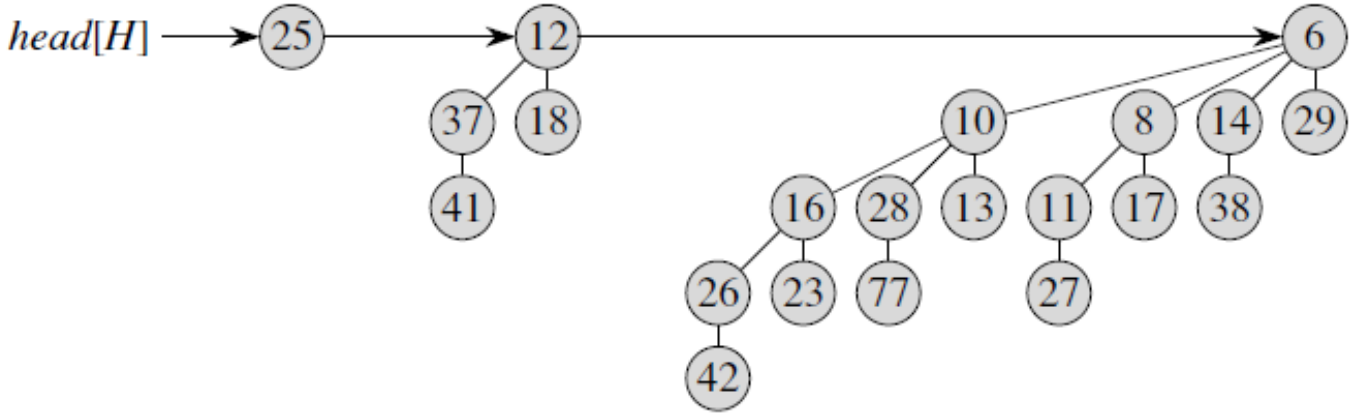






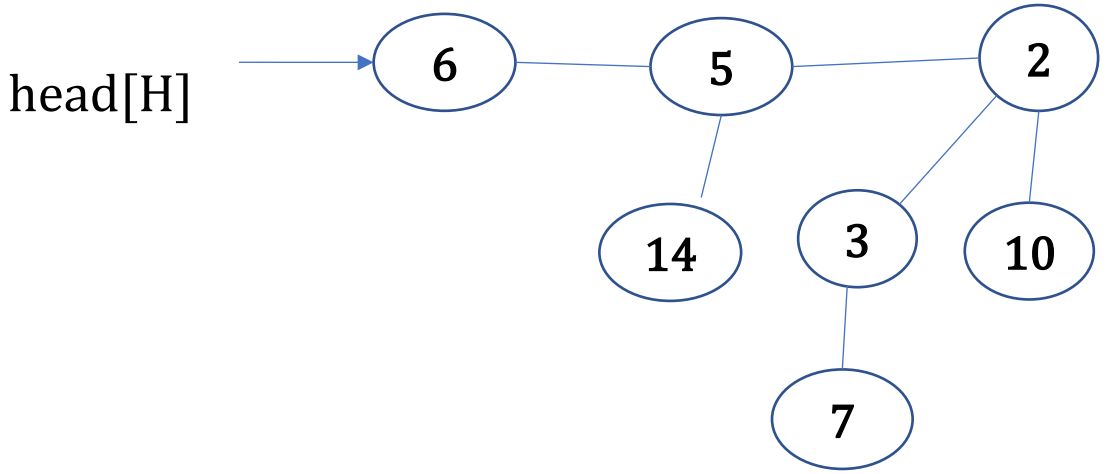
=

دو هرم حاصل شده را باهم اجتماع می کنیم و نتیجه بصورت زیر می شود.

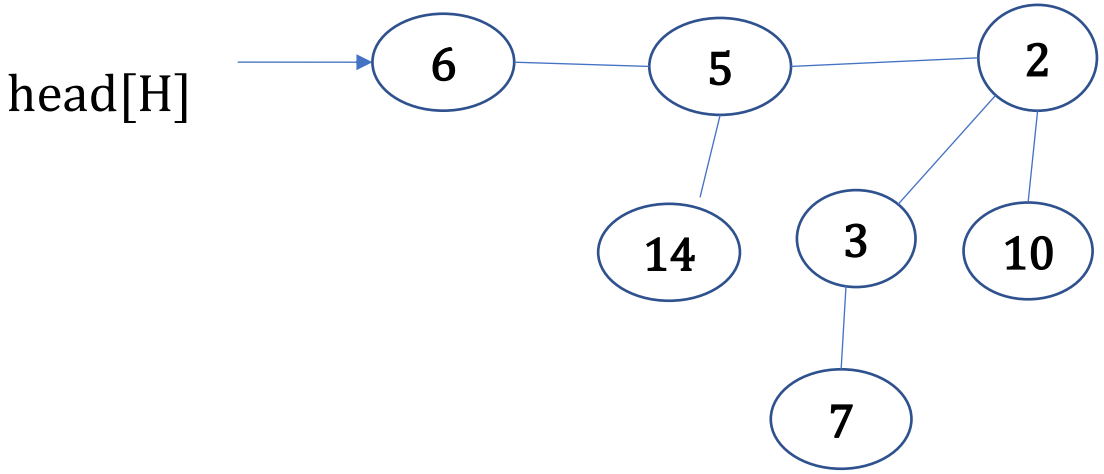


نکته: زمان اجرای عمل استخراج گره با کمترین مقدار از هرم دوجمله ای برابر است با $O(logn)$

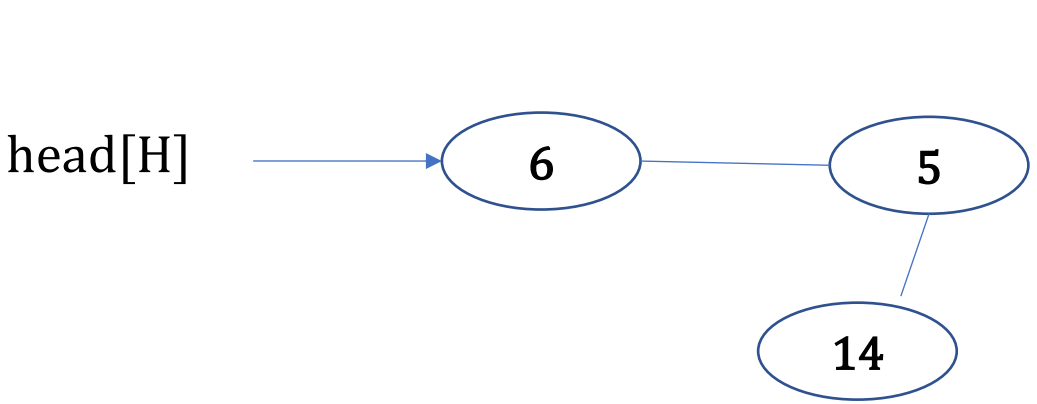
کار در کلاس: مراحل استخراج گره با کمترین مقدار را در هرم زیر بنویسید.

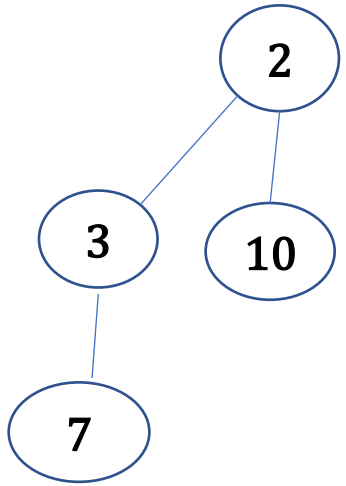


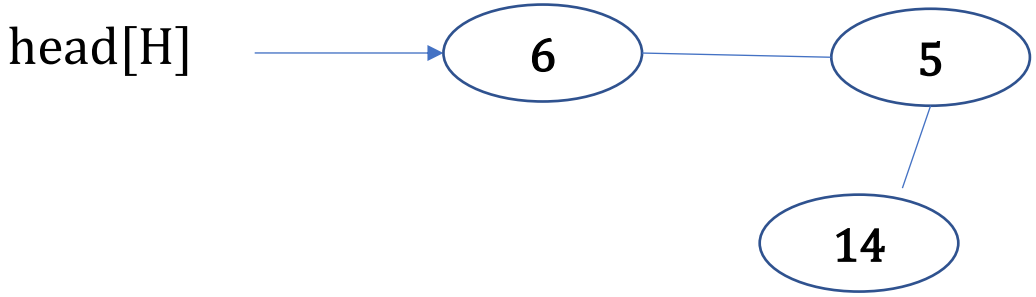
جواب: کوچکترین کلیدها در ریشه درختان دوجمله ای هستند که در اینجا گره با عدد ۲ می باشد.

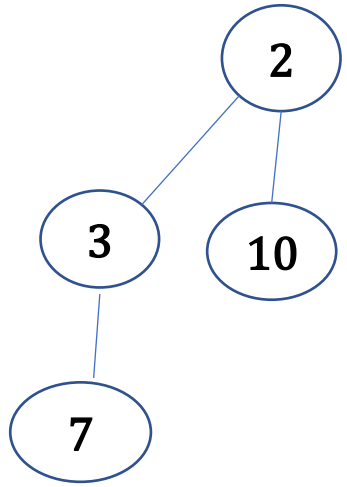


درخت دوجمله ای
مربوط به گره ۲ را از
هرم جدا می کنیم.

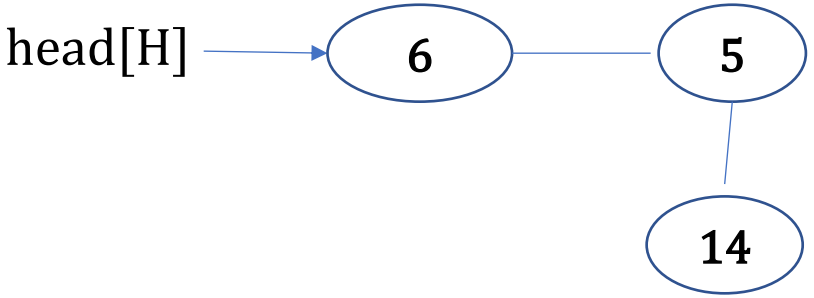


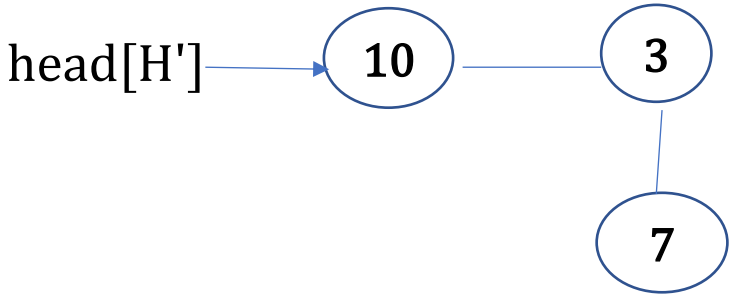


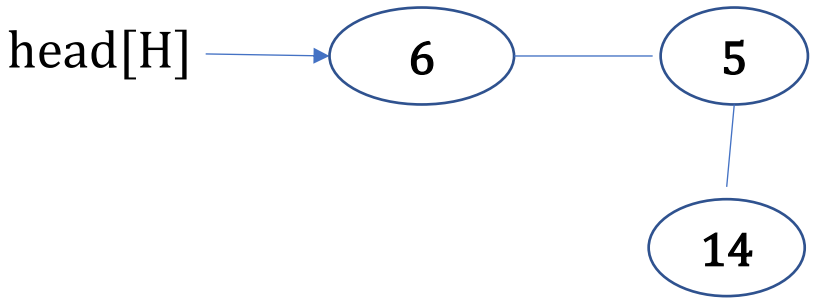


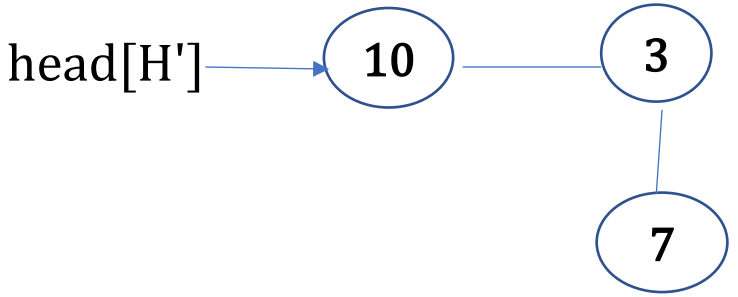


گره مربوط به عدد ۲ را حذف کرده و با فرزندان آن یک هرم جدید می سازیم. معمولا چون درختان دوجمله ای از درجه های کمتر شروع می شوند بنابراین ترتیب فرزندان را معکوس می کنیم.

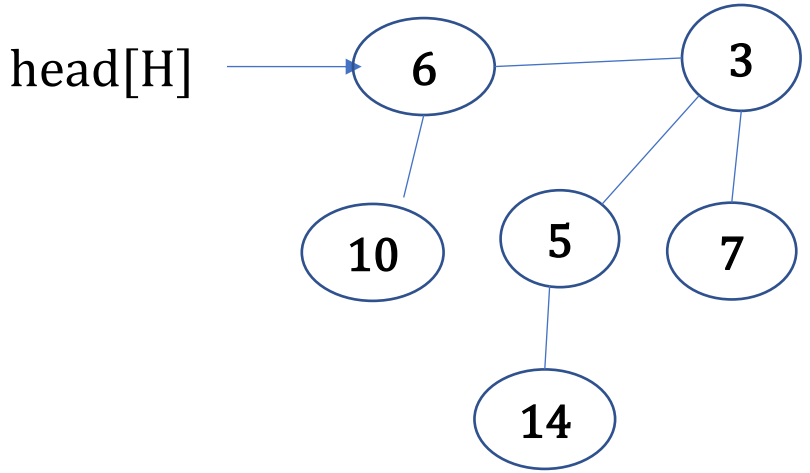








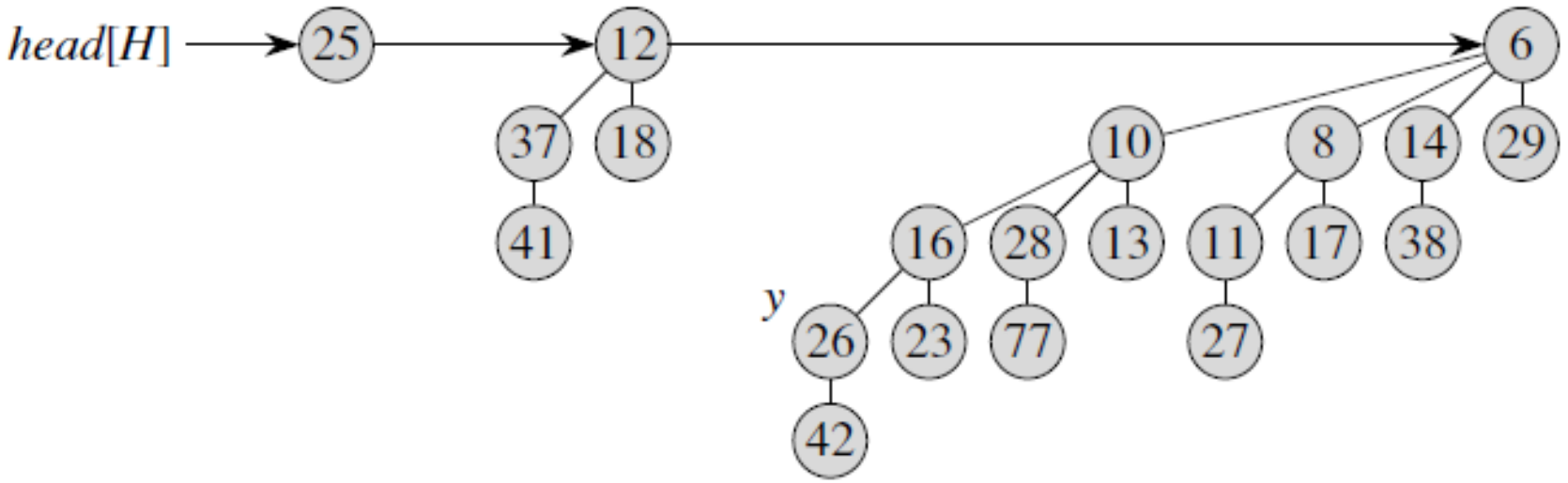
دو هرم حاصل شده را باهم اجتماع می کنیم و نتیجه بصورت زیر می شود.

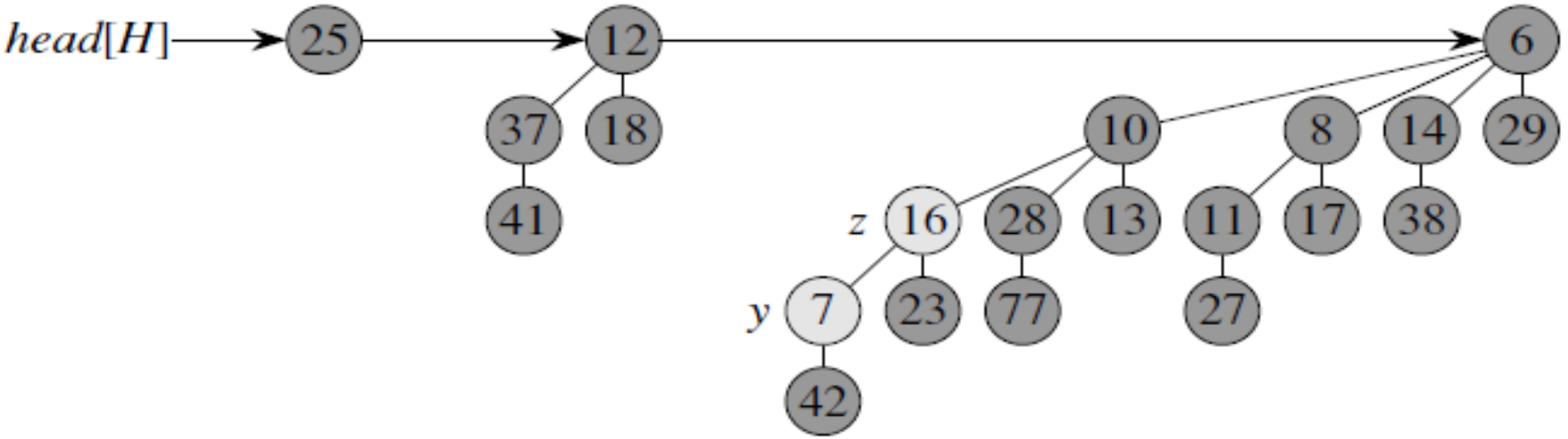


۵- کاهش یک کلید از هرم دوجمله ای

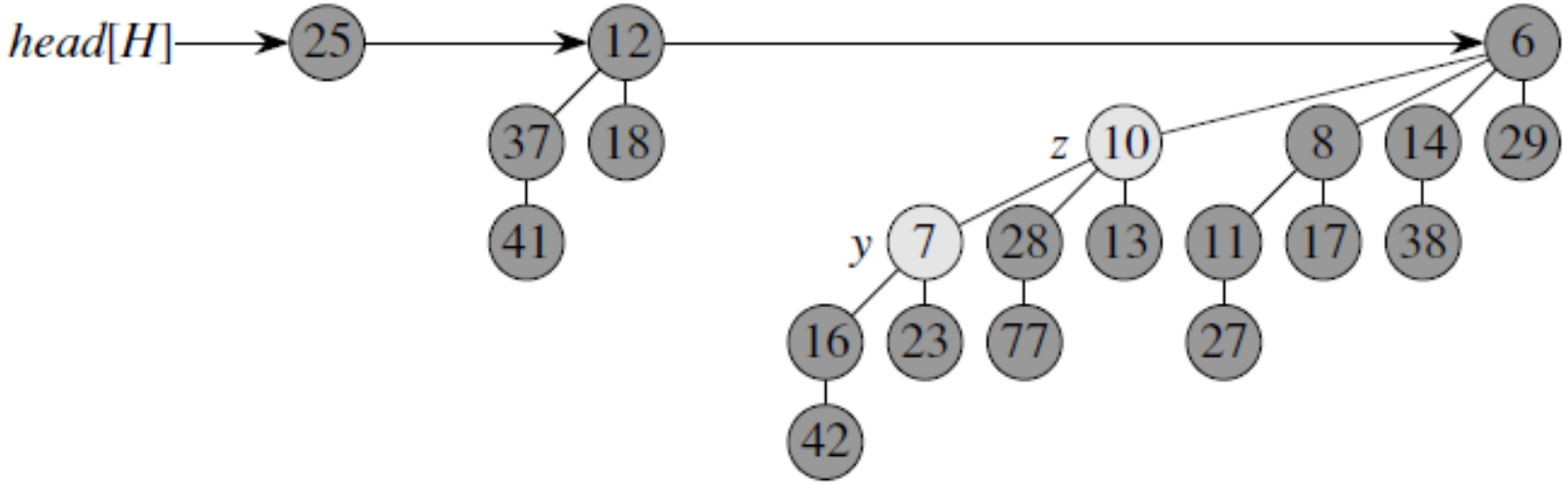
این عمل مقدار کلید x را در هرم دوجمله ای به مقدار جدید k کاهش می دهد (k<x). پس از قرار دادن مقدار k، آن را در هرم به سمت ریشه حرکت می دهد تا وقتیکه از والدش بزرگتر باشد (چون هر درخت دوجمله ای بصورت MinHeap است).

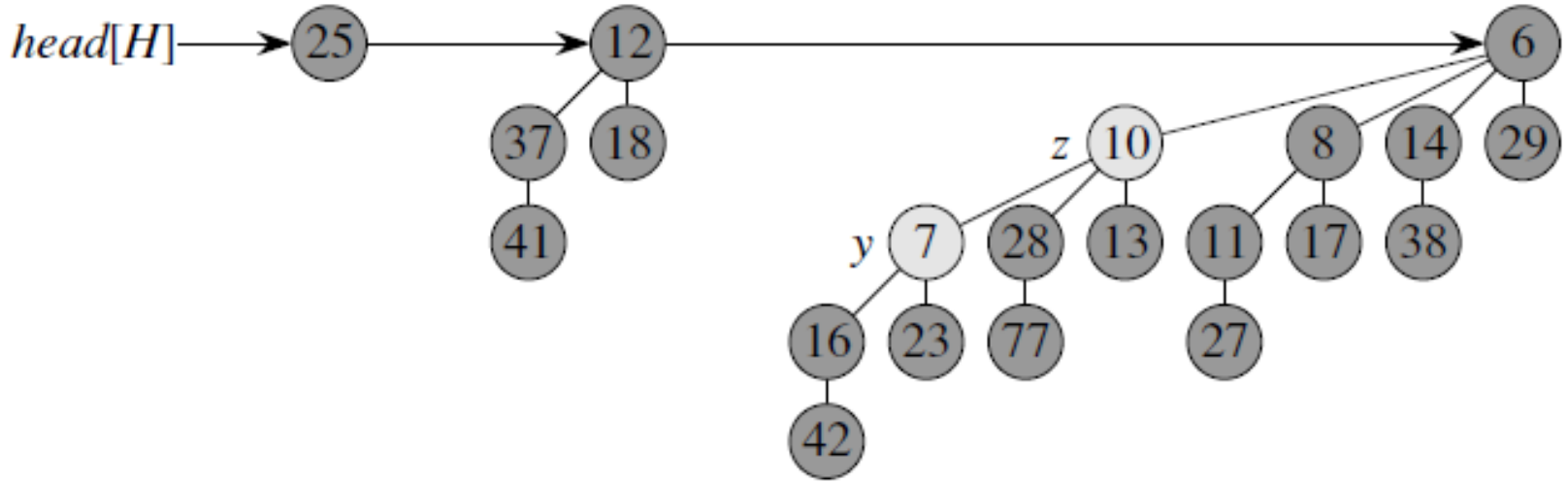
بعنوان مثال: مراحل کاهش مقدار گره y را به عدد ۷ بنویسید.



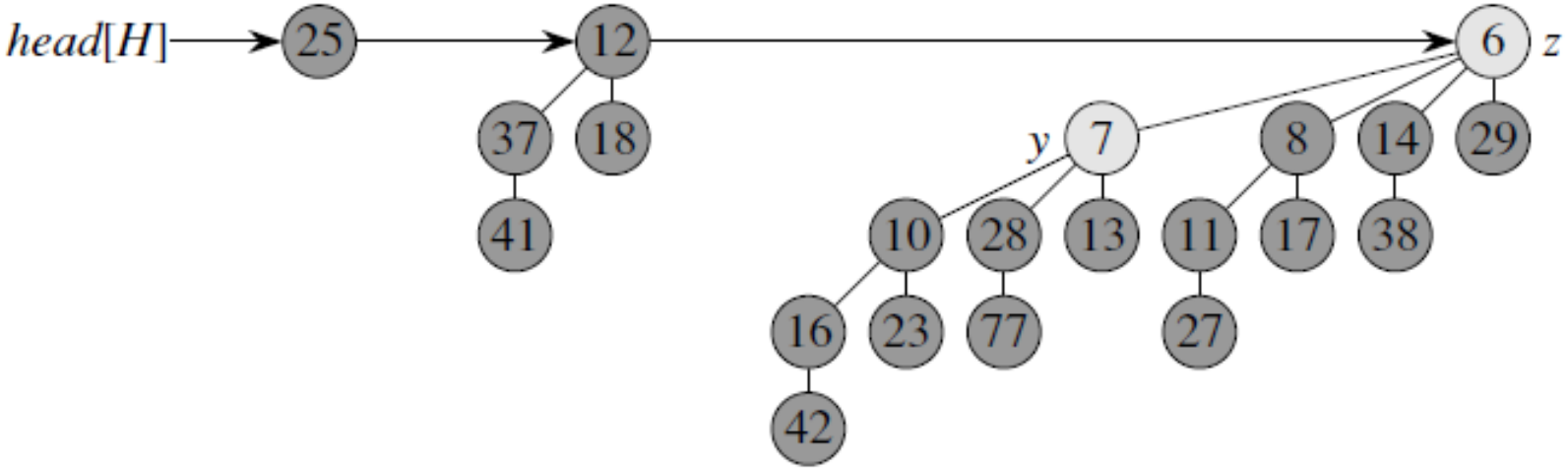


چون مقدار گره y کمتر از مقدار گره z است پس جابجا می شوند.





چون مقدار گره y کمتر از مقدار گره z است پس جابجا می شوند.



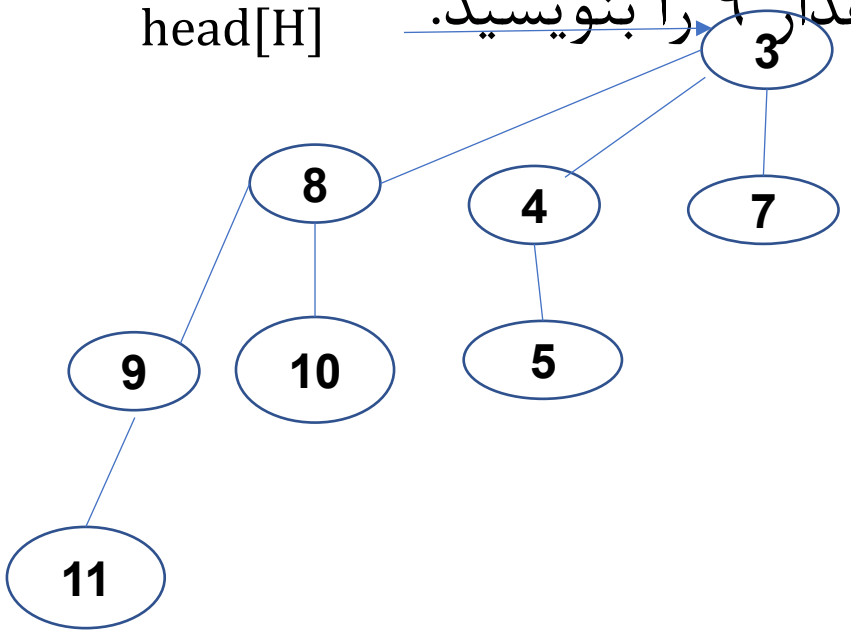
نکته: زمان اجرای عمل کاهش مقدار یک کلید در هرم برابر است با $O(logn)$

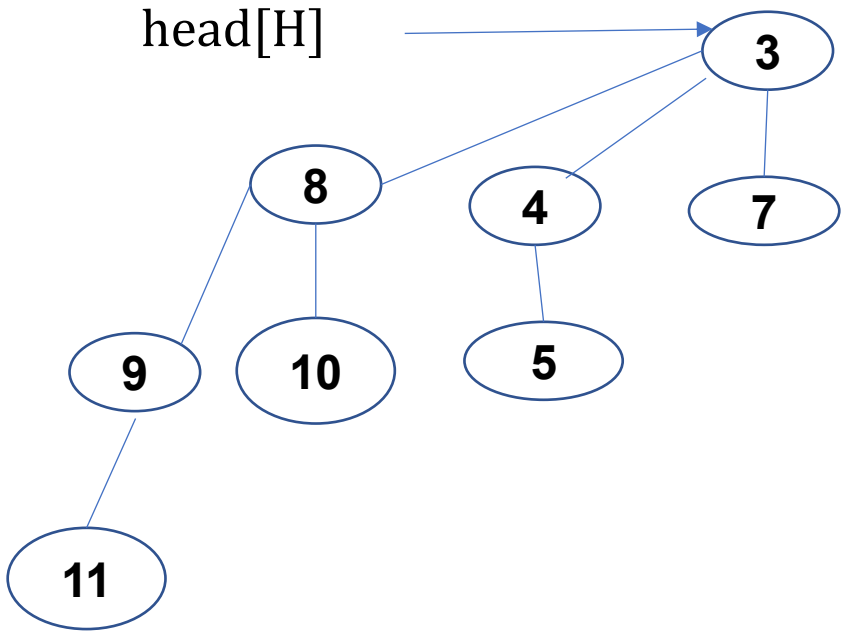
۶- حذف کلید X از هرم دوجمله ای

برای اینکار، مقدار کلید X را در هرم دوجمله ای به مقدار جدید ∞- کاهش می دهد پس از قرار دادن مقدار ∞- ، آن را در هرم به سمت ریشه حرکت می دهد که حتماً به ریشه خواهد رسید (چون از همه کمتر است). سپس با روش استخراج کوچکترین مقدار از هرم، آن را حذف می کند.

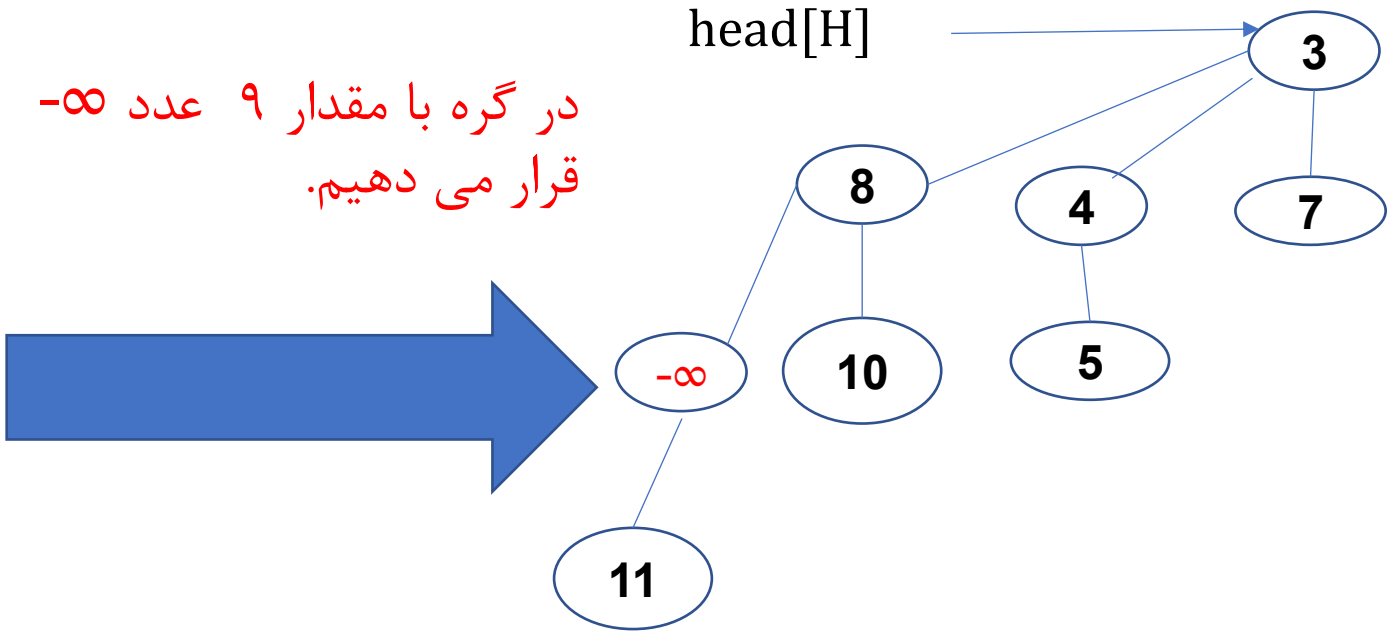
نکته: زمان اجرای عمل حذف یک گره در هرم برابر است با $O(logn)$

بعنوان مثال: مراحل حذف گره با مقدار ۹ را بنویسید.

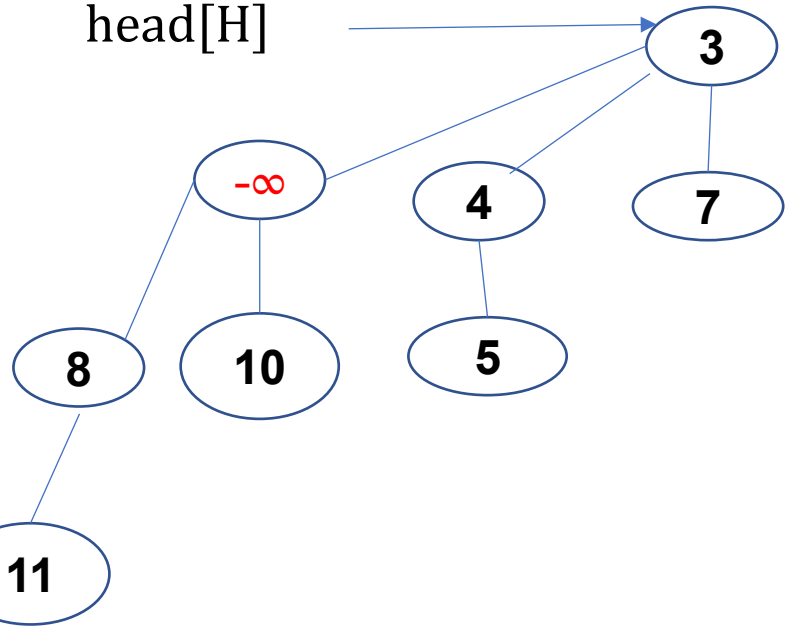




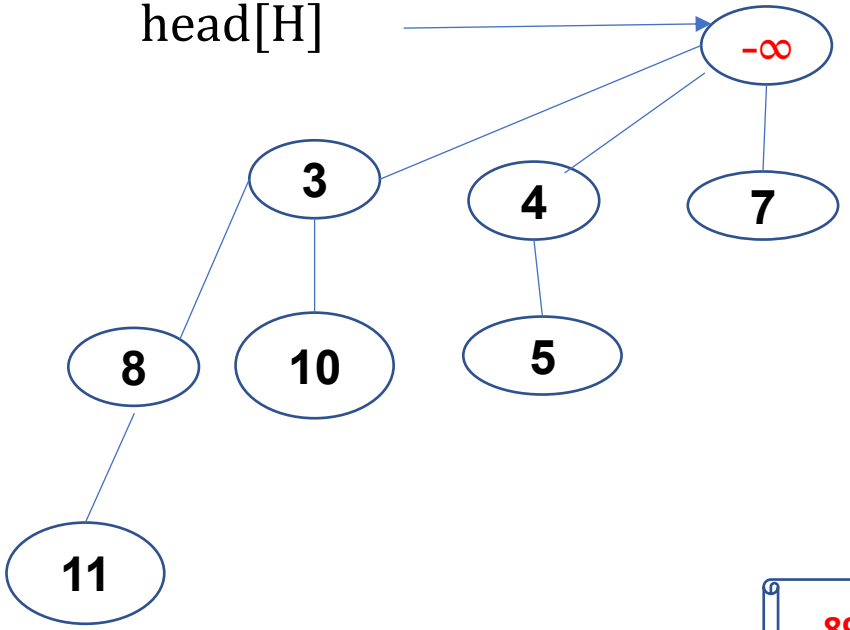
در گره با مقدار ۹ عدد ∞- قرار می دهیم.

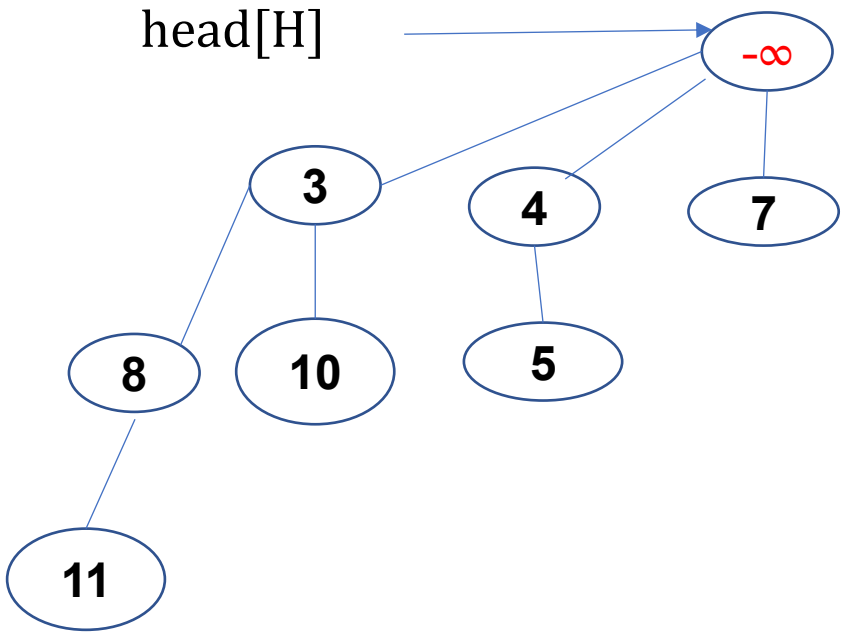


گره با مقدار ∞- به سمت ریشه حرکت می کند.



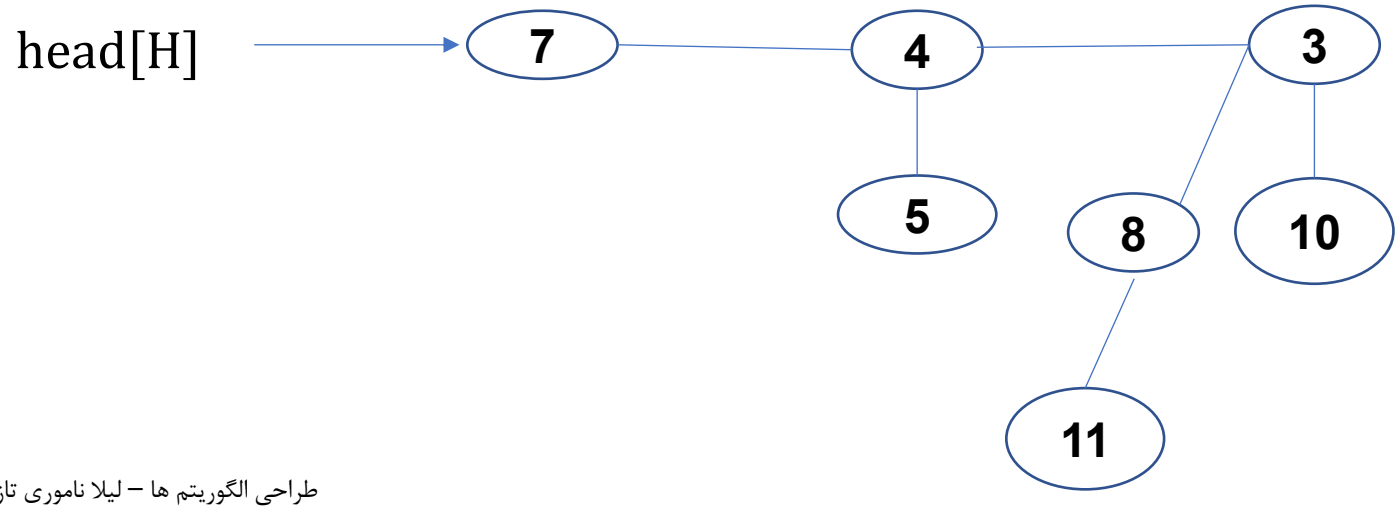
گره با مقدار ∞- به سمت ریشه حرکت می کند.





با روش استخراج کوچکترین مقدار از هرم، گره با مقدار ∞- را حذف می کند.

گره مربوط به عدد ∞- را حذف کرده و با فرزندان آن یک هرم جدید می سازیم. معمولا چون درختان دوجمله ای از درجه های کمتر شروع می شوند بنابراین ترتیب فرزندان را معکوس می کنیم.



تمرینات سری ۲

۱- با عناصر زیر یک هرم دوجمله ای بسازید.

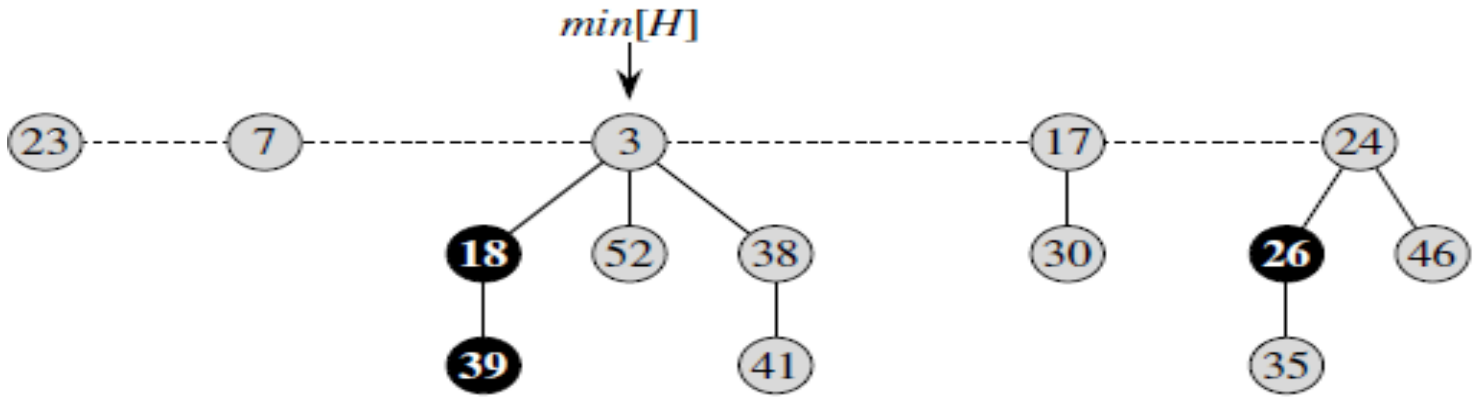
3,7,16,9,4,6,13,1,10,5,11,15,12

۲- مراحل استخراج گره با کوچکترین کلید را در هرم ایجاد شده در تمرین ۱ بنویسید (۱۲ مرتبه)

۳- مقدار گره با کلید ۱۲ را در هرم ایجاد شده در تمرین ۱ را به عدد ۲ تغییر دهید.

۴- مراحل حذف گره با مقدار ۴ را در هرم ایجاد شده در تمرین ۱ بنویسید.

- ✓ تعریف: هرم فیبوناچی، مشابه هرم دوجمله ای، شامل مجموعه ای از درخت های MinHeap است یعنی مقدار گره والد از مقادیر گره های فرزندان کوچکتر یا مساوی است.
- ✓ هر کدام از درخت ها می توانند نامرتب باشند یعنی ترتیب فرزندان مهم نیست.
- ✓ درخت های MinHeap لزوماً، درخت دوجمله ای نیست (برعکس هرم دوجمله ای که درخت ها حتما باید دوجمله ای باشند)
- ✓ درخت های MinHeap لزوماً برحسب درجه ریشه ها مرتب شده نیست (برعکس هرم دوجمله ای که درخت ها باید برحسب درجه ریشه ها مرتب شده باشند)
- ✓ معمولاً،

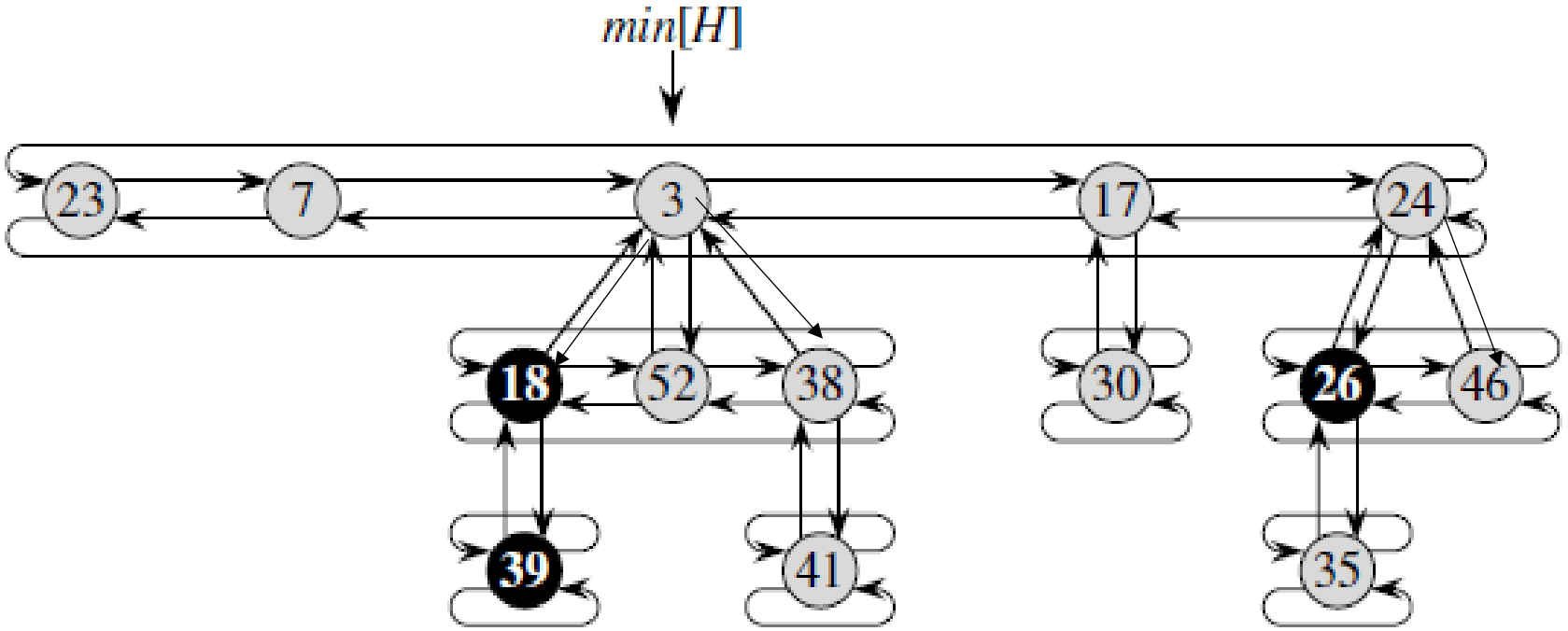


✓ نمایش (ذخیره) هرم های فیبوناچی:

- هر گره x شامل یک اشاره گر p[x] به والدش، و یک اشاره گر child[x] به هر یک از فرزندانش می باشد.

- فرزندان گره x بصورت لیست دوپیوندی حلقوی بهم متصل می باشند. هر فرزند بنام y دارای دو اشاره گر چپ (left) و راست (right) می باشد که بترتیب به همزادهای چپ و راست y اشاره می کنند.

- مثال:



✓ هر گره، علاوه بر اشاره گره های گفته شده در اسلاید قبلی، دارای دو فیلد دیگر نیز است:

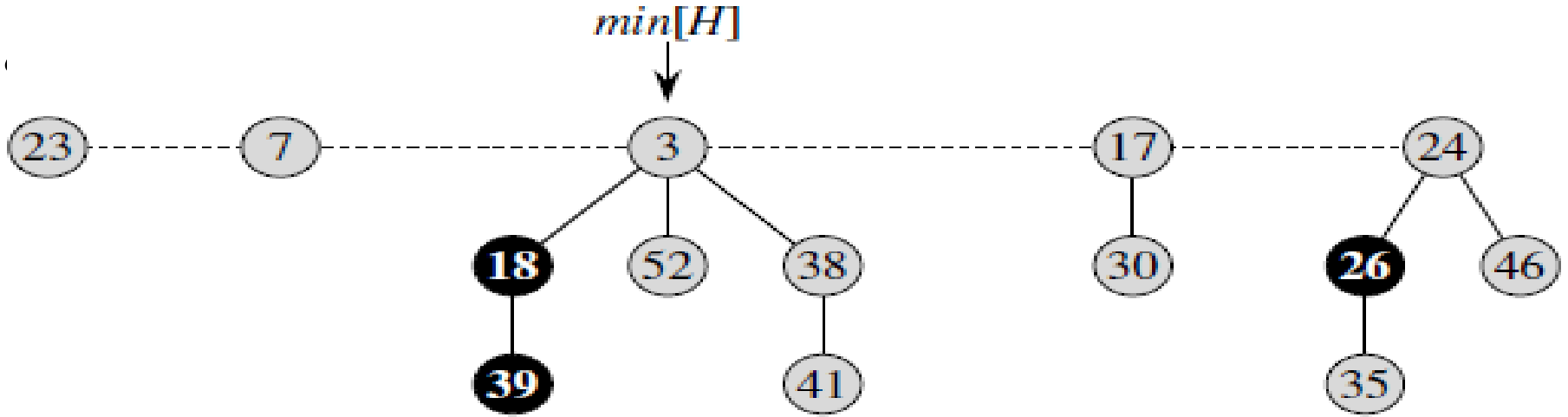
۱- degree[x] : درجه گره x را نگهداری می کند.

۲-mark[x]: یک مقدار منطقی را نشان می دهد و مشخص می کند که آیا گره x از آخرین باری که فرزند گره دیگری شده است فرزندی را از دست داده است؟ گره های جدیداً ایجاد شده بدون علامت هستند (False) و گره x زمانی که فرزند گره دیگری می شود بدون علامت می گردد (False)

- مثال

توپر سیاه نشان

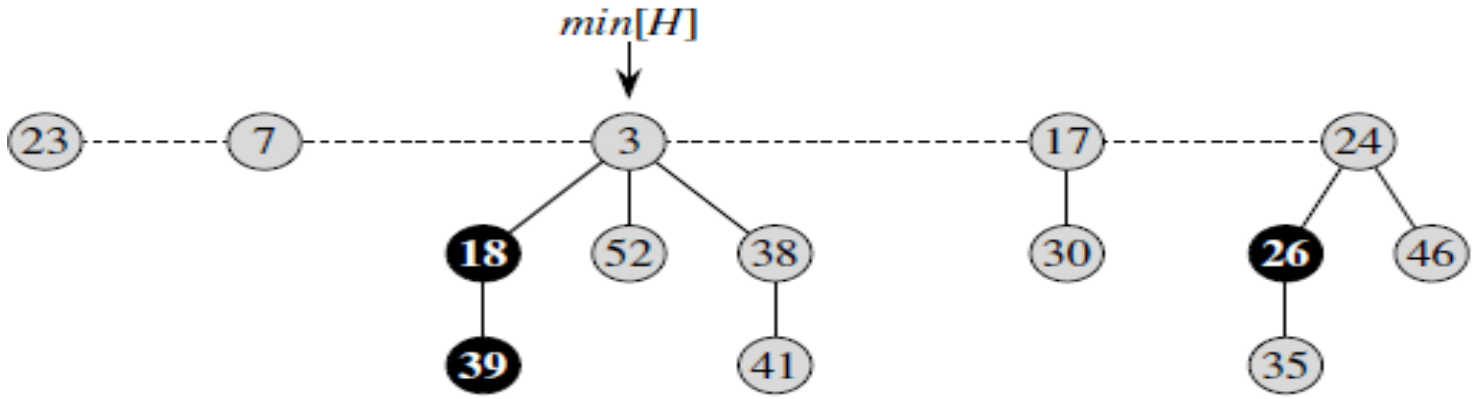
داده



✓ تابع پتانسیل (که برای تحلیل سرشکن شده عملگرها استفاده خواهد شد) برای هرم فیبوناچی H بصورت زیر محاسبه می شود : $Q(H) = t(H) + 2m(H)$

که در آن، $t(H)$ تعداد درختان هرم و $m(H)$ تعداد گره های علامتدار را مشخص می کنند.

- مثال: در شکل زیر $m(H)=3$, $t(H)=5$ بنابراین $Q(H)=5+2*3=11$ خواهد بود.



- $n(H)$ تعداد کل گره های هرم را نشان می دهد. در مثال بالایی داریم: $n(H)=14$

- $D(n)$ بیشترین درجه هرم با n گره را مشخص می کند و همیشه رابطه زیر برقرار است:

$$D(n) \leq \lfloor \log n \rfloor \rightarrow D(14) \leq \lfloor \log 14 \rfloor \leq 3$$

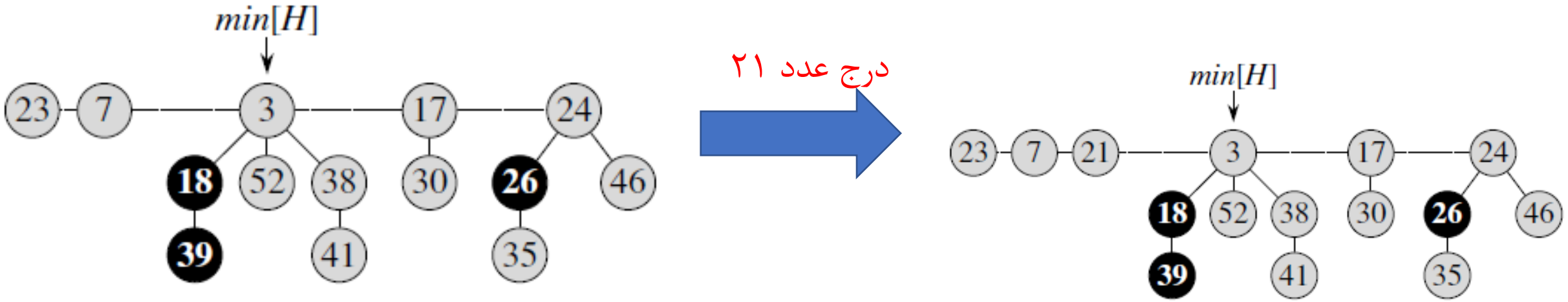
۱- ایجاد هرم فیبوناچی جدید و خالی

برای اینکار، یک شی هرم فیبوناچی H را ایجاد می کنیم بطوریکه

- $n[H]=0$
- $min[H]=NULL$ چون در H، هیچ درختی وجود ندارد.
- $t(H)=0$, $m(H)=0$ بنابراین $Q(H)=0$ یعنی پتانسیل هرم فیبوناچی خالی برابر صفر است در نتیجه هزینه سرشکن شده برابر هزینه واقعی آن یعنی $O(1)$ است.

۲- درج گره x به هرم فیبوناچی

- گره x به لیست ریشه ها و سمت چپ گره (همزاد چپ) با مقدار min[H] اضافه می شود.
- اگر x<min[H] آنگاه min[H] به گره جدید x اشاره خواهد کرد.
- n[H]=n[H]+1



چون پتانسیل هرم طبق فرمول ($Q(H) = t(H) + 2m(H)$) فقط یک واحد تغییر کرد ؟؟؟ بنابراین هزینه سرشکن شده طبق فرمول $\hat{c_i} = c_i + Q(D_i) - Q(D_{i-1})$ برابر خواهد بود با هزینه واقعی بعلاوه یک واحد یعنی $O(1)+1$ که همان $O(1)$ است.

۳- واحد سازی (اجتماع: Union) دو هرم H1 و H2 و تشکیل یک هرم جدید H

- ریشه های هرم ها بهم وصل می شوند و ساختار هرم ها عوض نمی شوند.
- $n[H]=n[H1]+n[H2]$ تعداد کل گره ها
- $t[H]=t[H1]+t[H2]$ تعداد درختان هرم
- $m[H]=m[H1]+m[H2]$ تعداد گره های علامتدار
- اختلاف پتانسیل هرم H با هرم های H1 و H2 برابر صفر است.

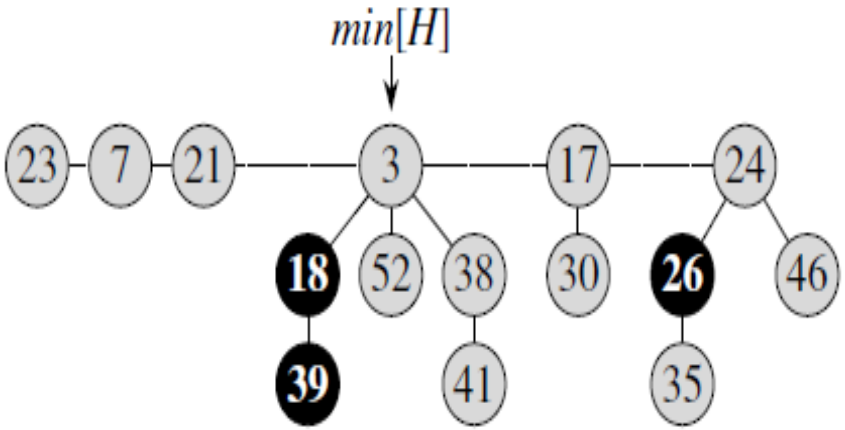
$$Q(H) - (Q(H1) + Q(H2))$$
$$= (t(H) + 2m(H)) - ((t(H1) + 2m(H1)) + (t(H2) + 2m(H2)))$$
$$= 0$$

چون اختلاف پتانسیل هرم برابر صفر شد بنابراین هزینه سرشکن شده برابر هزینه واقعی آن یعنی (O(1) است.

۴- استخراج گره با کمترین مقدار

این عمل، گره با کوچکترین مقدار را پیدا کرده و آن را از هرم حذف می کند.

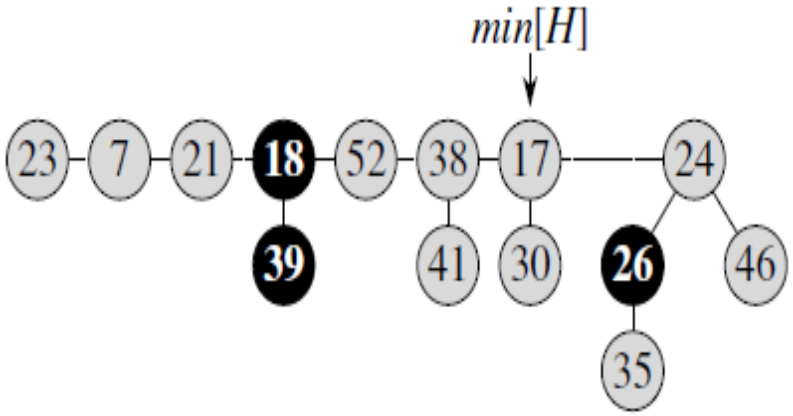
بعنوان مثال: مراحل استخراج گره با کمترین مقدار را در هرم زیر بنویسید.

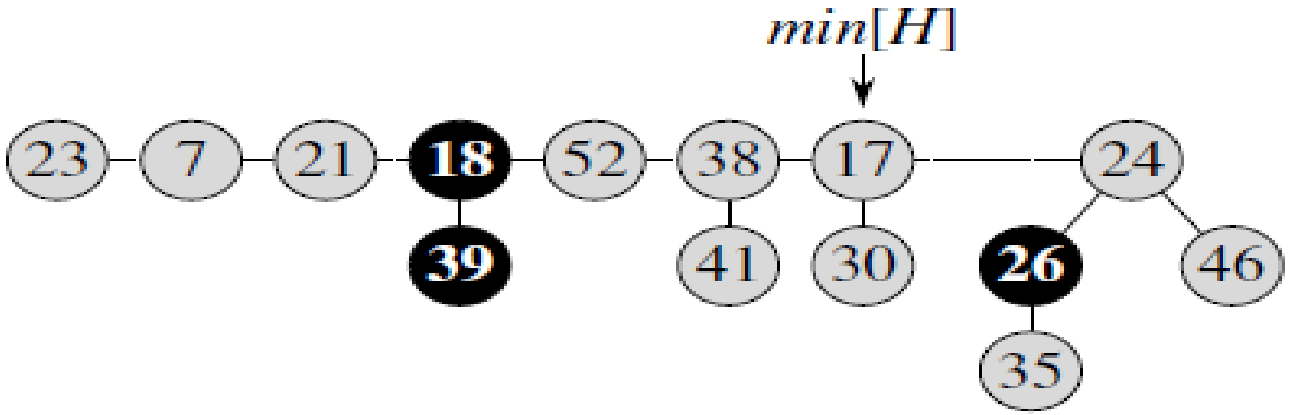


مرحله ۱: حذف گره با مقدار ۳

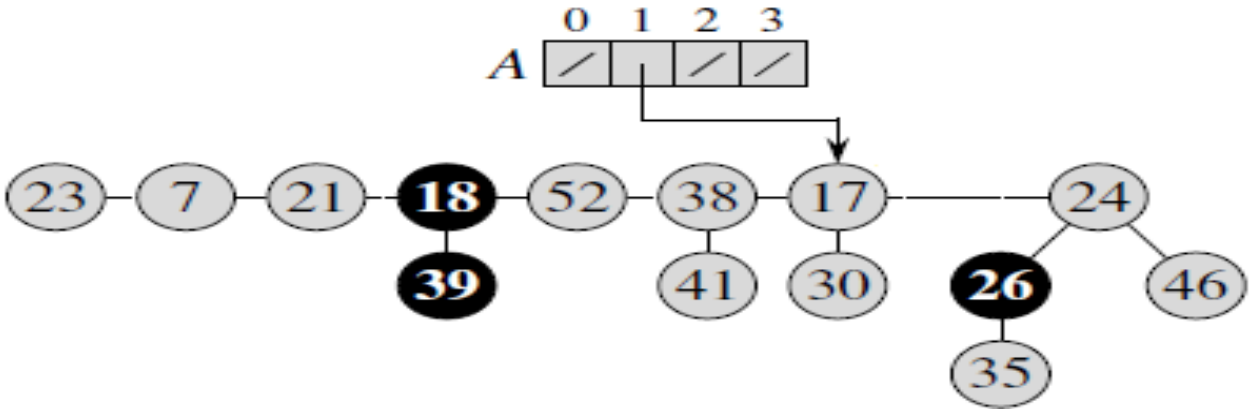
۱- فرزندان گره حذف شده بعنوان ریشه در نظر گرفته می شوند.

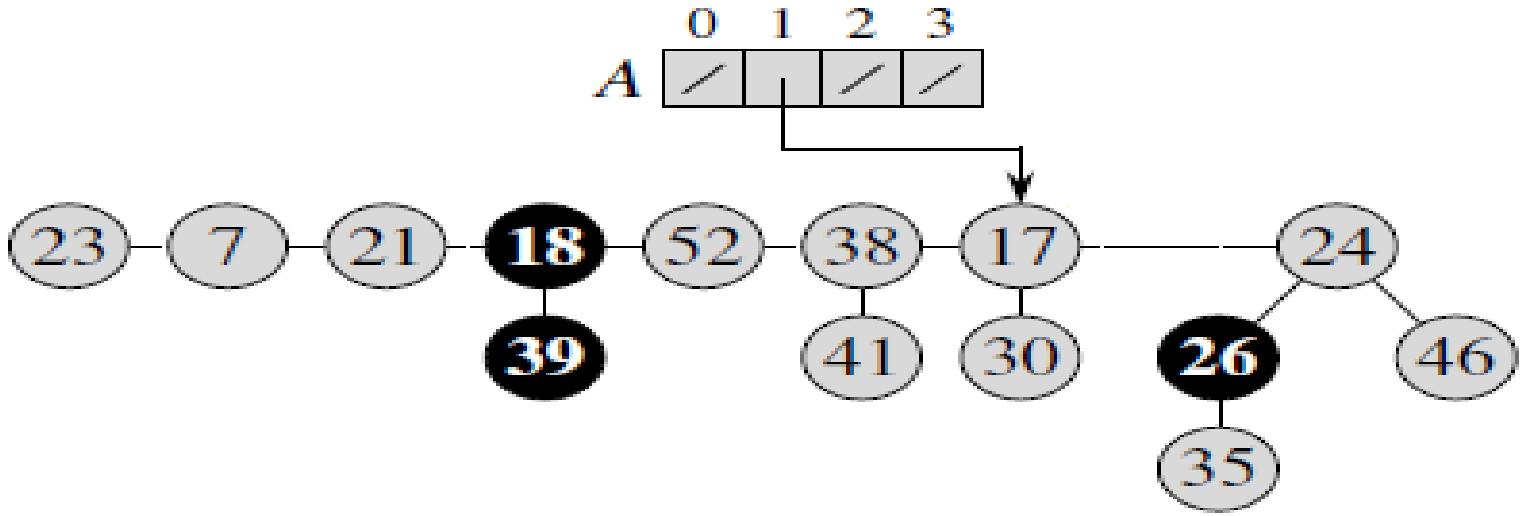
۲- مقدار اشاره گر min[H] به همزاد راست گره حذف شده اشاره خواهد کرد.



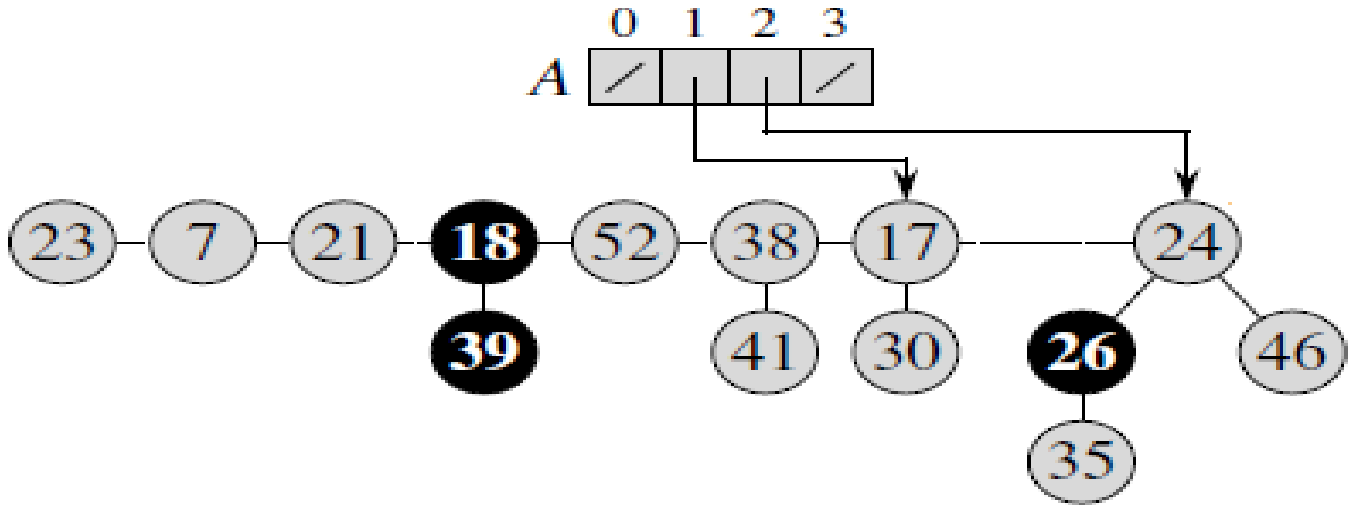


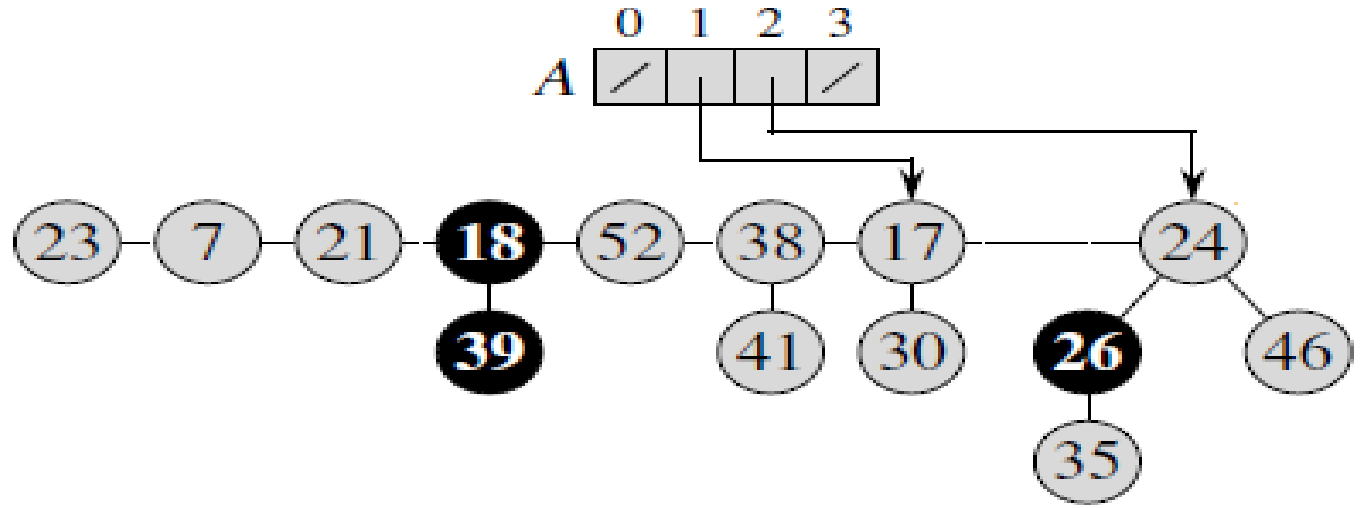
مرحله ۲: یک آرایه کمکی A بطول حداکثر درجه بعلاوه یک واحد ($D(n)$) و داریم $D(n) \leq \lfloor \log n \rfloor$ در این مثال داریم $n=15$ (تعداد قبل از حذف) و $D=3$ خواهد بود) برای اشاره کردن به گره های ریشه در نظر می گیریم. از گره $\min[H]$ شروع کرده و $A[i]$ به ریشه ای اشاره خواهد کرد که درجه آن برابر i باشد. $A[1]$ به ریشه ۱۷ اشاره می



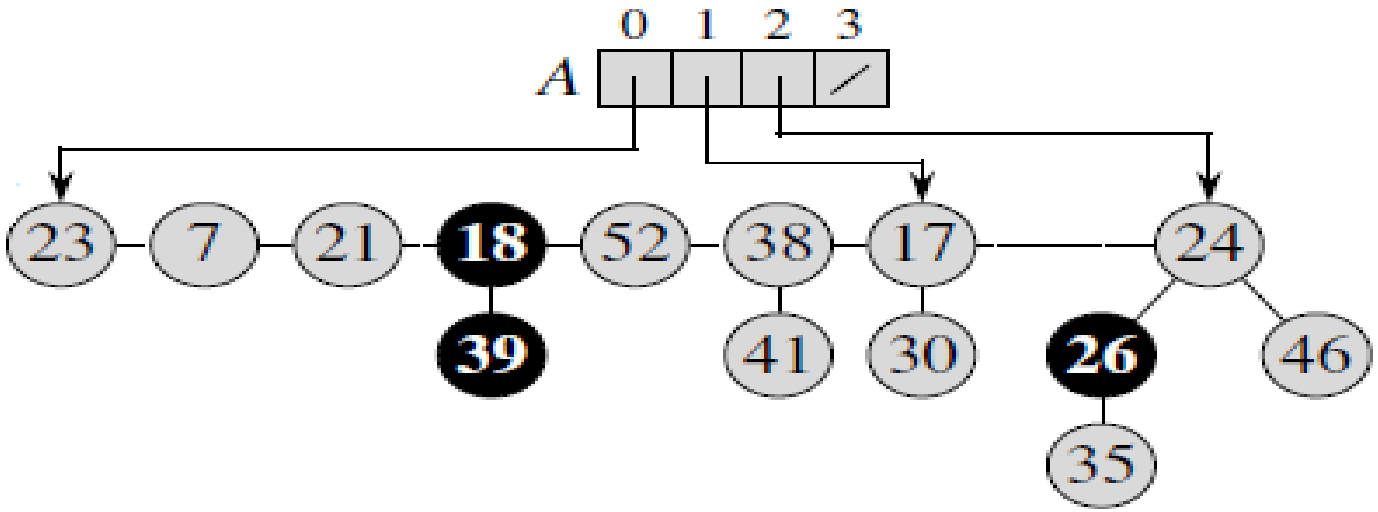


A[2] به ریشه ۲۴ اشاره می کند چون درجه آن برابر ۲ است.

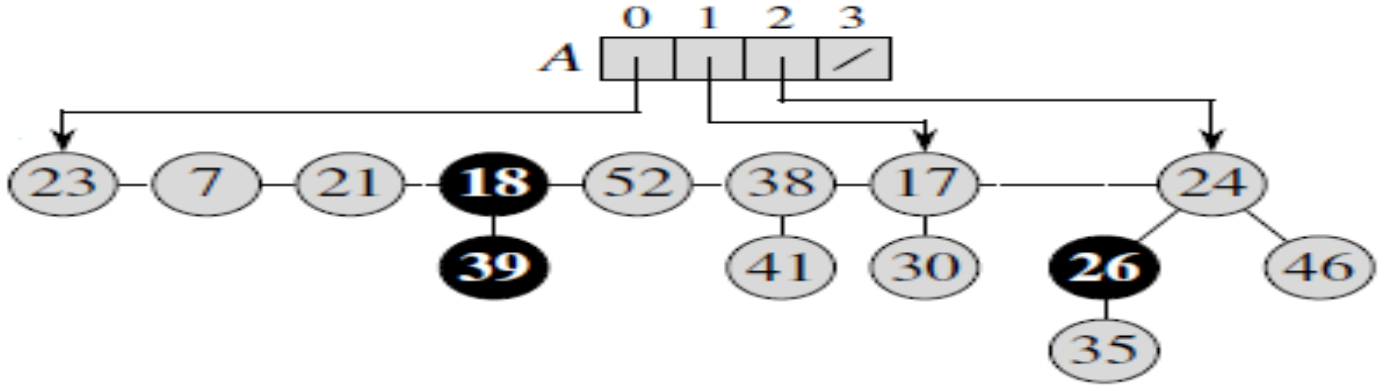




A[0] به ریشه 23 اشاره می کند چون درجه آن برابر صفر است.



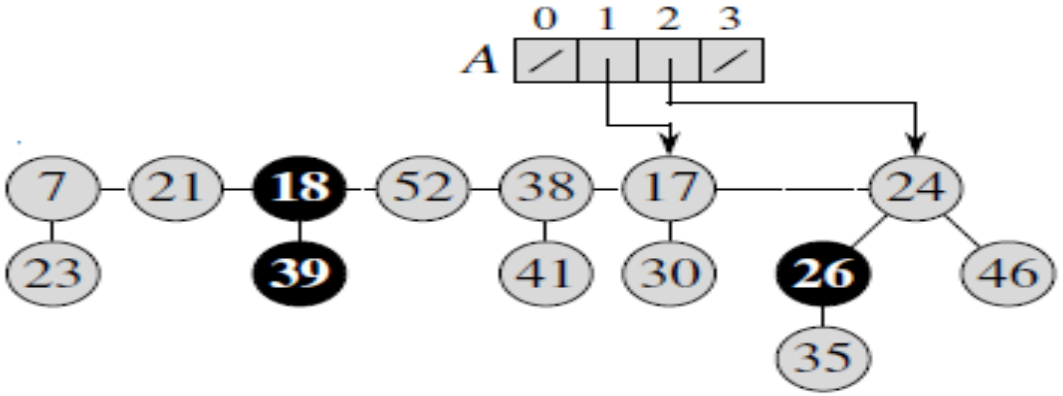
A[3] خالی می ماند چون ریشه با درجه ۳ وجود ندارد.

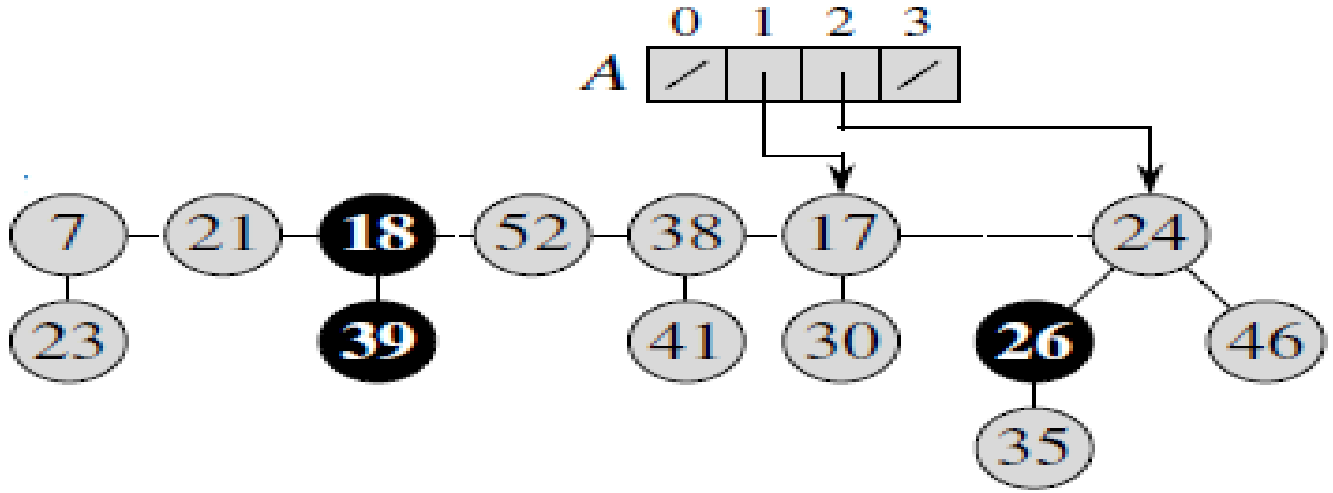


مرحله ۳: وقتی همه عناصر آرایه A مشخص شدند (بجز بعضی عناصر) از ابتدای آرایه A شروع کرده و اولین عنصری که NULL نیست را با درخت موجود بعدی که هم درجه هستند (یا درخت اخیراً ترکیب شده) ترکیب می کنیم. و یک درخت دوجمله ای جدید ایجاد می کنیم.

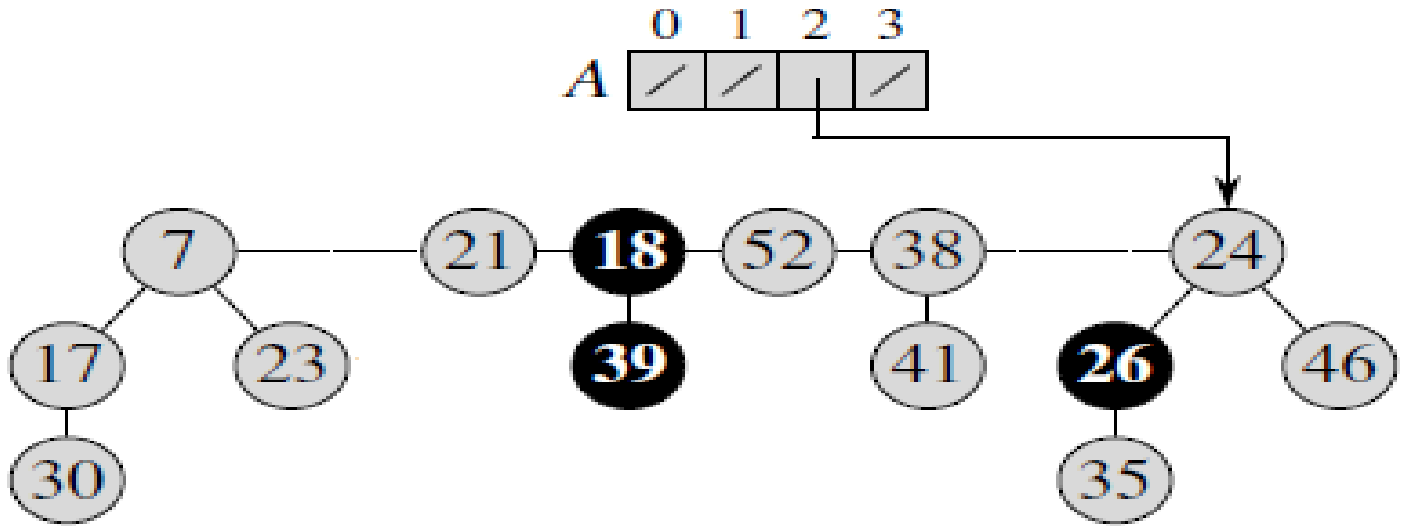
A[0] که به ریشه ۲۳ اشاره می کند را با ریشه ۷ که هم درجه هستند ترکیب می کنیم.

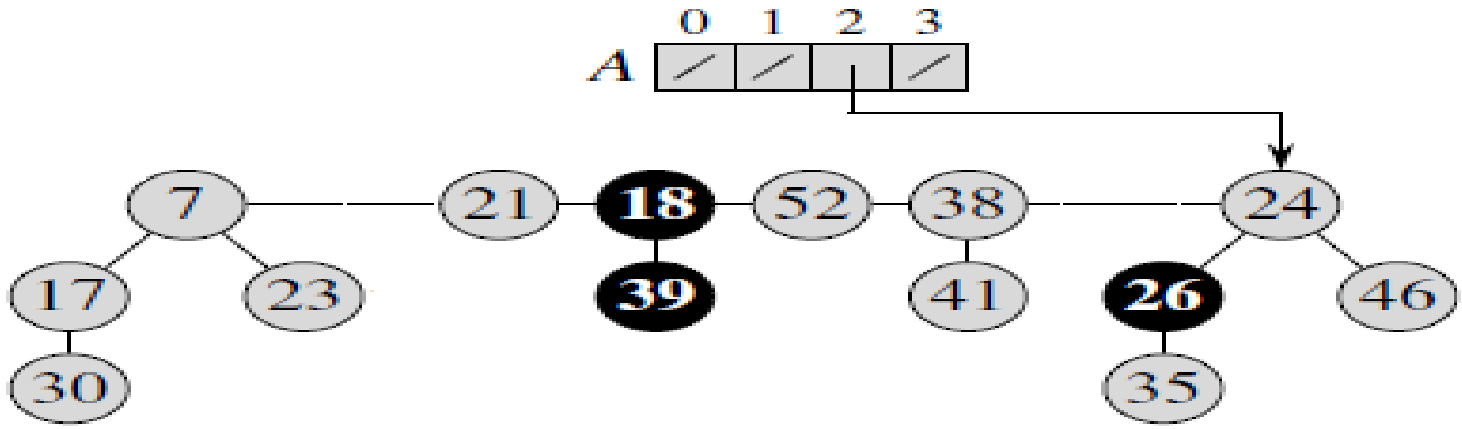
نکته: وقتی گره y به فرزند گره x تبدیل می شود علامت (mark) گره y برابر False می شود.



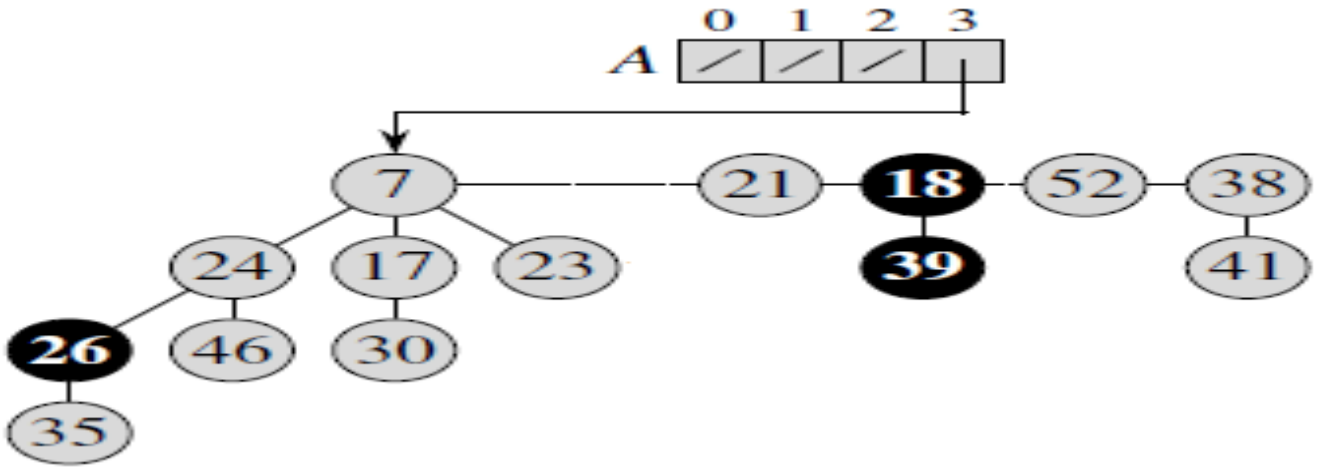


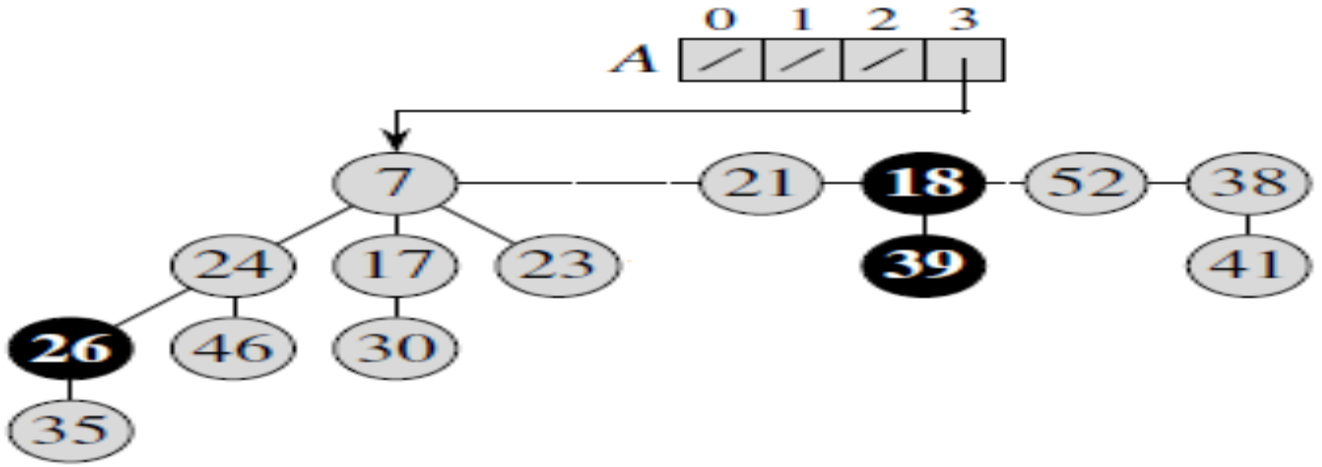
A[1] که به ریشه ۱۷ اشاره می کند را با درخت اخیراً ترکیب شده یعنی ۷، ترکیب می کنیم.



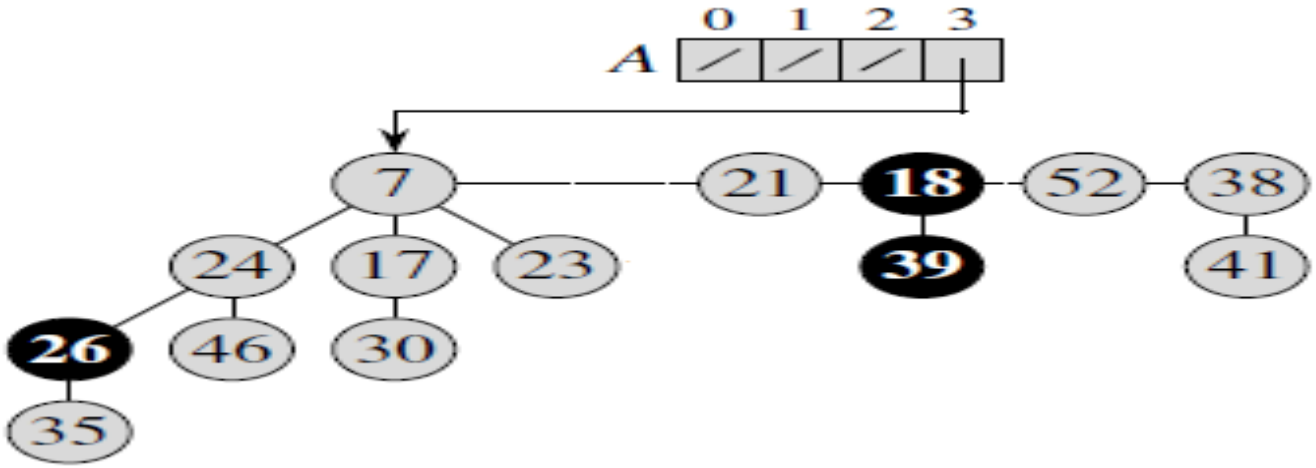


A[2] که به ریشه 24 اشاره می کند را با درخت اخیراً ترکیب شده یعنی ۷ ترکیب می کنیم.

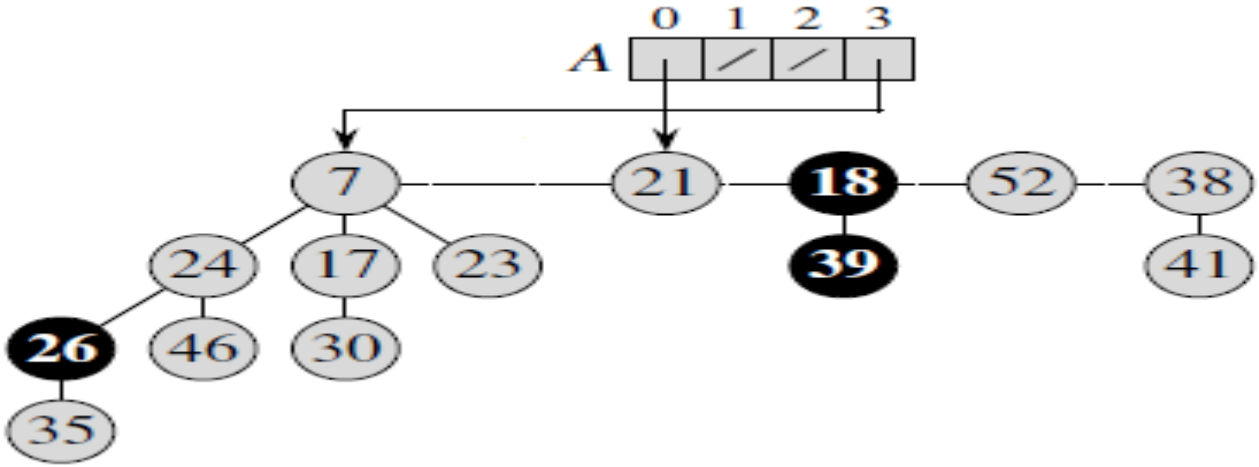


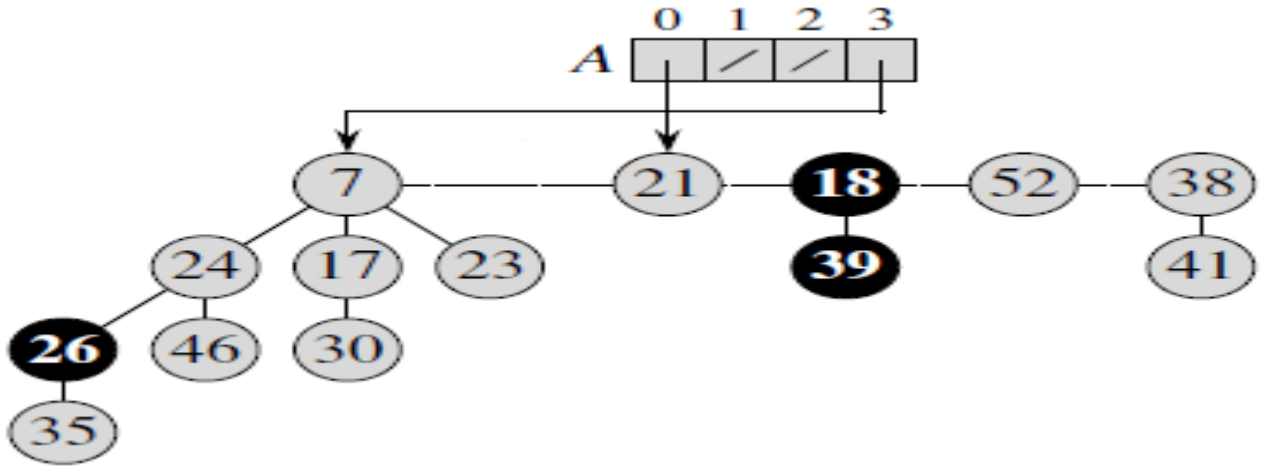


A[3] که به ریشه 7 اشاره می کند نمی تواند با هیچ ریشه ای ترکیب شود چون ریشه هم درجه ندارد

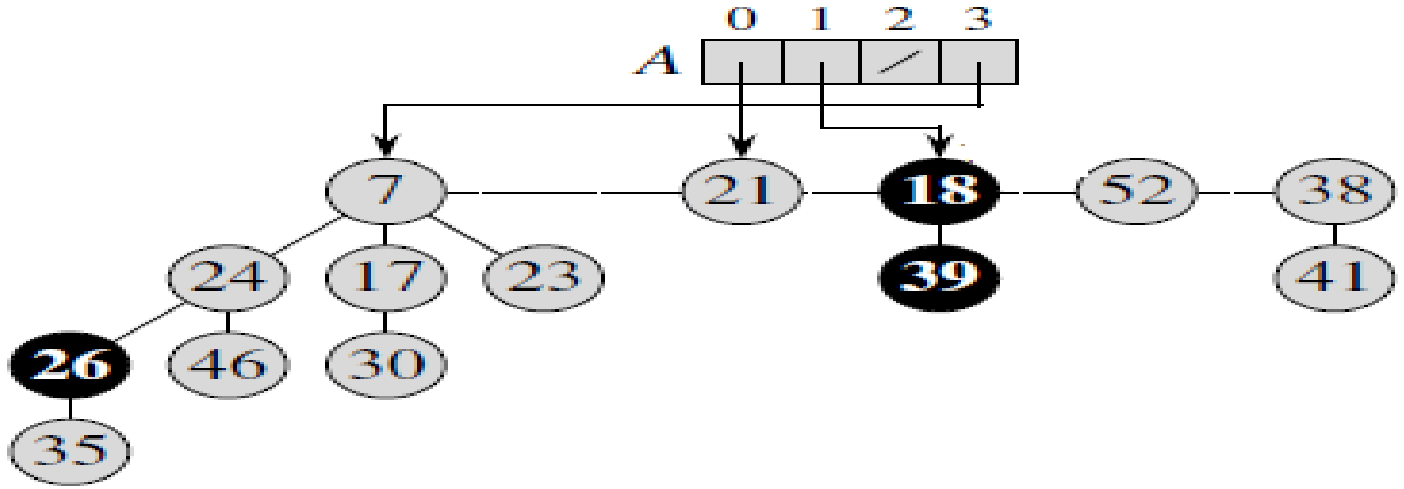


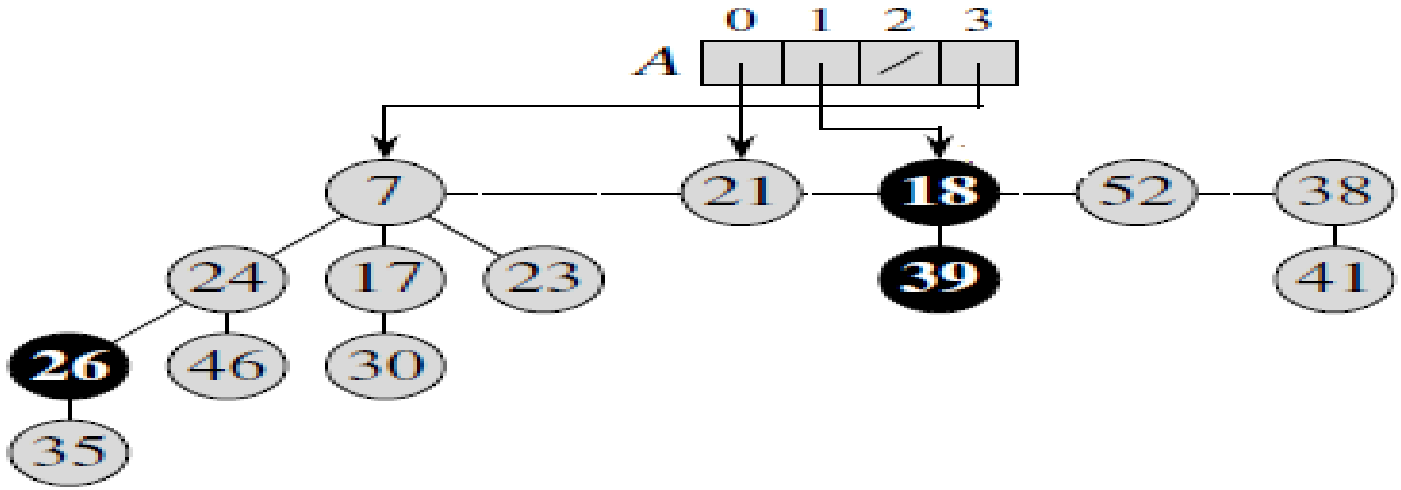
تکرار مرحله ۲: از ادامه درخت اخیراً ترکیب شده یعنی ۷، آرایه A را پر می کنیم. A[i] به ریشه ای اشاره خواهد کرد که درجه آن برابر i باشد.
A[0] به ریشه 21 اشاره می کند.



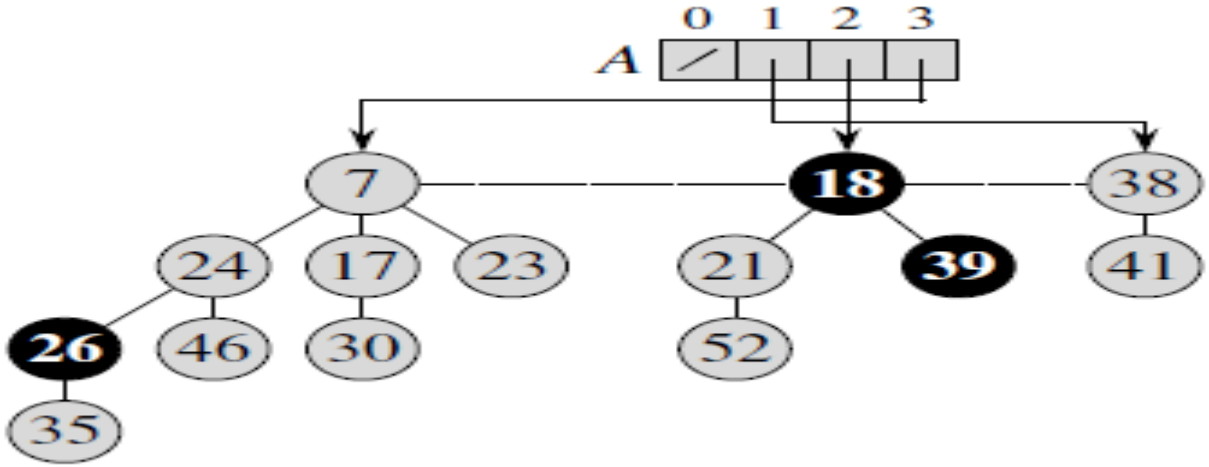


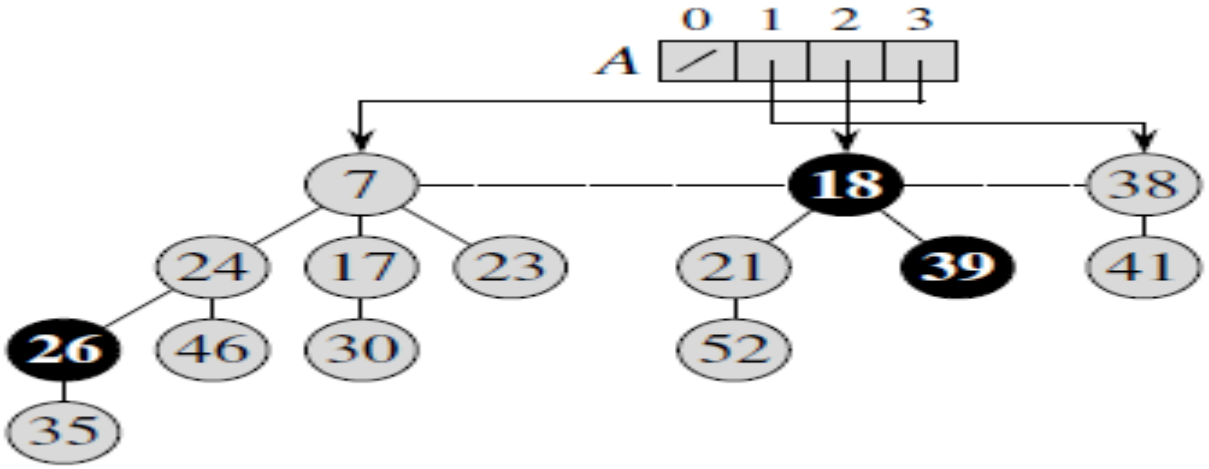
A[1] به ریشه 18 اشاره می کند.





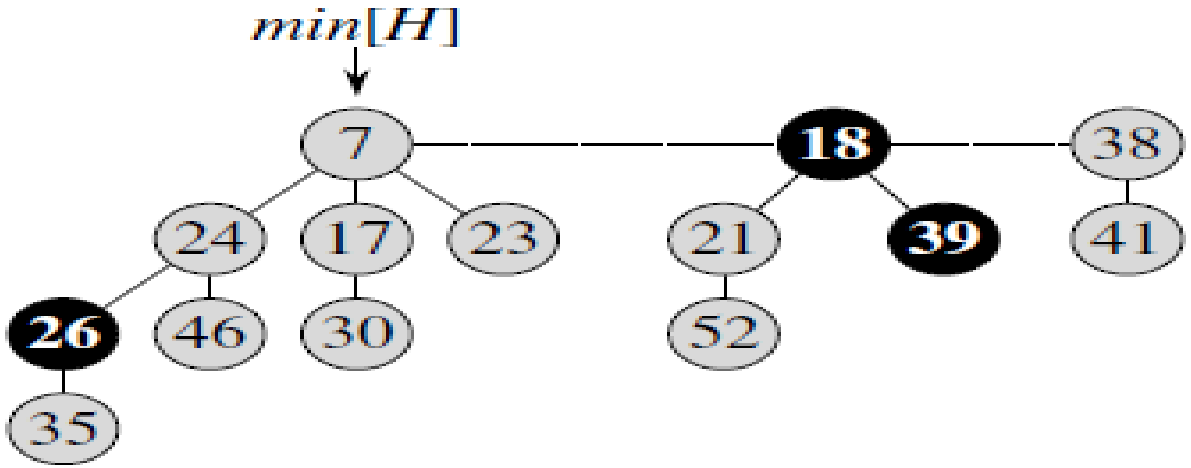
تکرار مرحله ۳: A[0] که به ریشه ۲۱ اشاره می کند را با ریشه ۵۲ که هم درجه هستند ترکیب می کنیم و سپس در نتیجه، A[1] که به ریشه ۱۸ اشاره می کند را با درخت اخیراً ترکیب شده یعنی ۲۱ ترکیب می کنیم





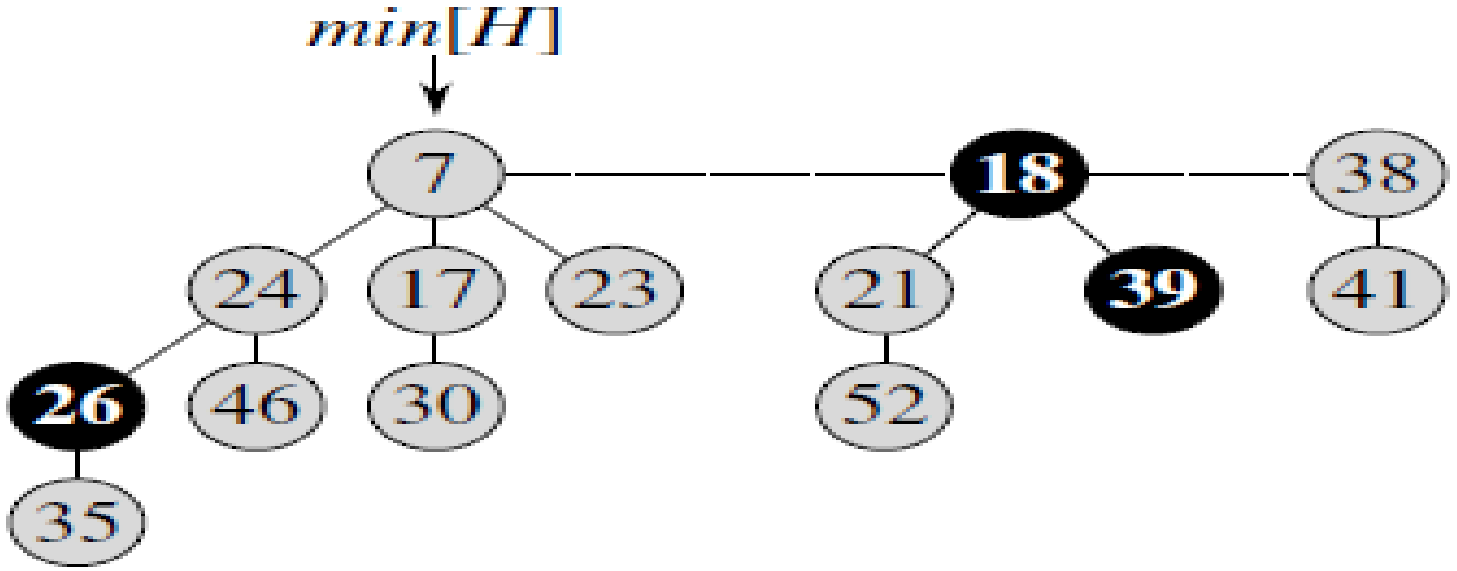
الگوریتم به اتمام می رسد چون دو ریشه هم درجه وجود ندارد.

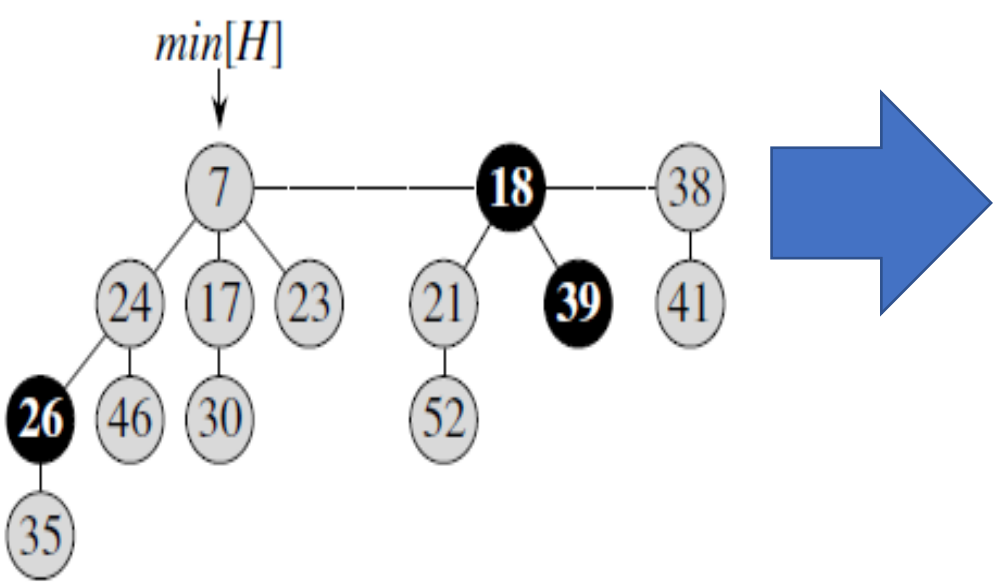
ضمنا min[H] به کمترین ریشه تنظیم می شود.

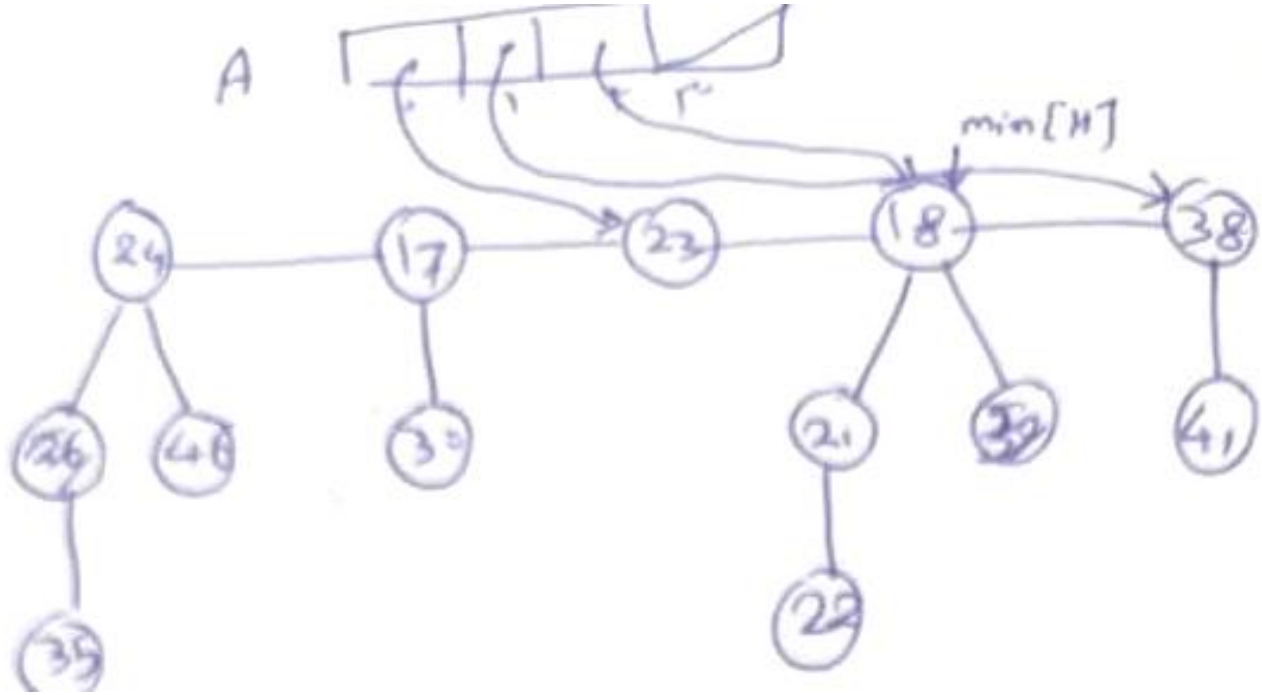


ثابت می شود که هزینه سرشکن شده برابر هزینه واقعی آن یعنی O(D(n)) یا O(logn) است.

کار در کلاس: مراحل استخراج گره با کمترین مقدار را در هرم زیر بنویسید.

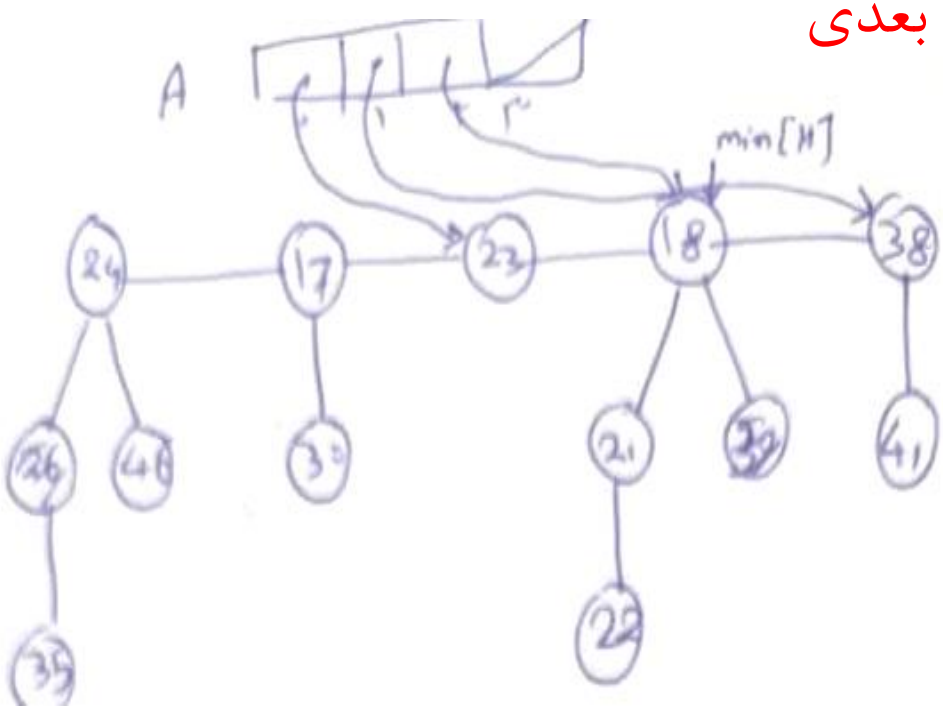


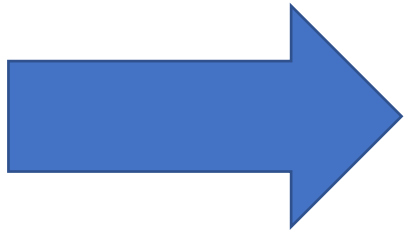


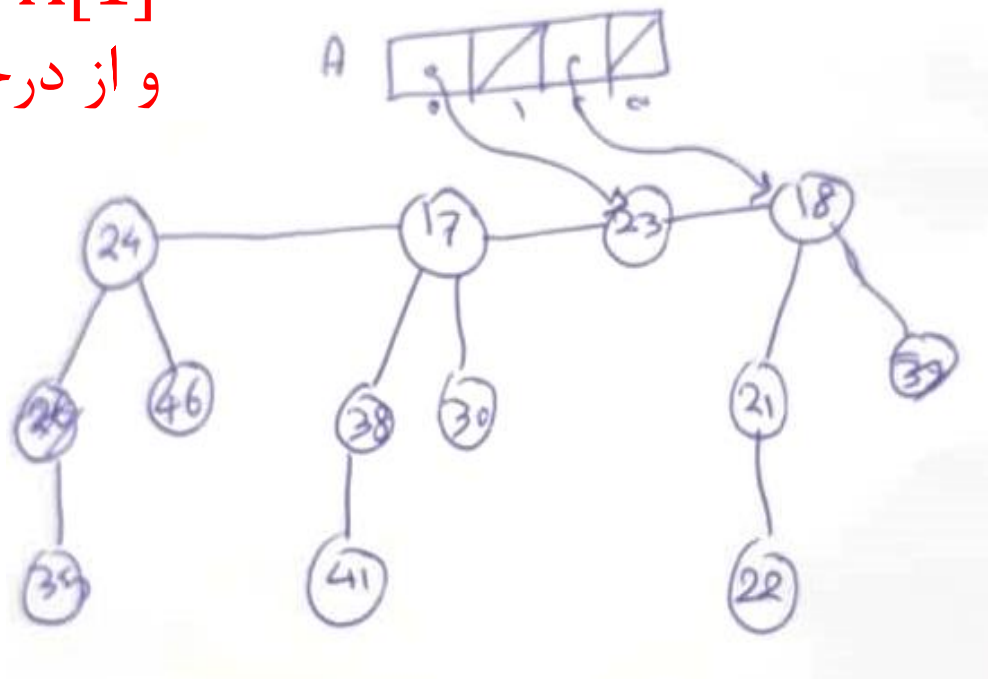


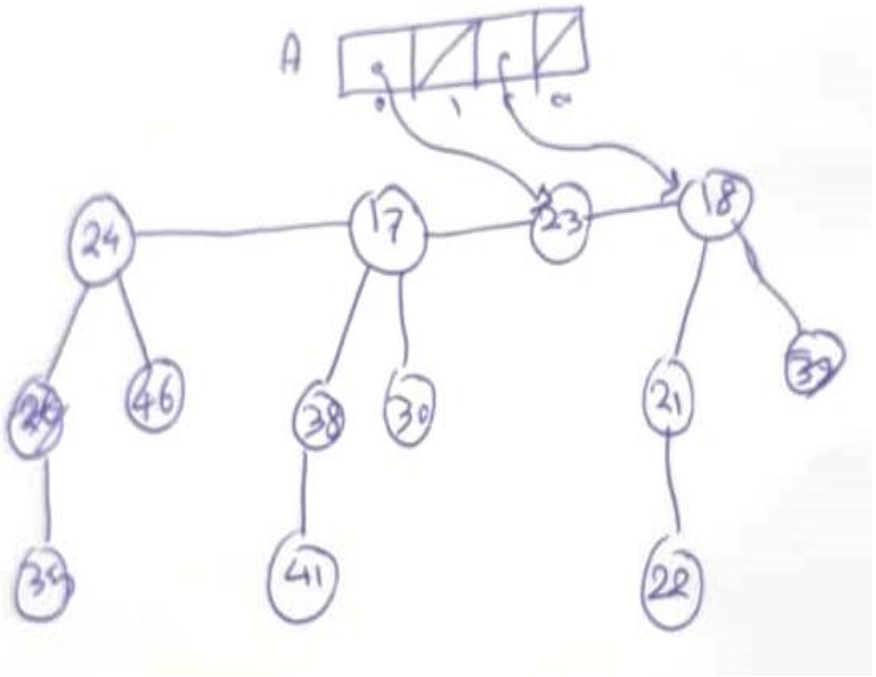
بعد از مشخص کردن عناصر آرایه A، از ابتدای آرایه A شروع کرده و اولین عنصری که NULL نیست را با درخت موجود بعدی که هم درجه هستند (یا درخت اخیراً ترکیب شده) ترکیب می کنیم. و یک درخت دوجمله ای جدید ایجاد می کنیم. چون در این هرم، فقط یک درخت دوجمله ای از درجه صفر وجود دارد بنابراین A[0] را نادیده گرفته و به سراغ A[1] می رود.

A[1] را با درخت دوجمله ای بعدی
و از درجه یک ترکیب می کند.

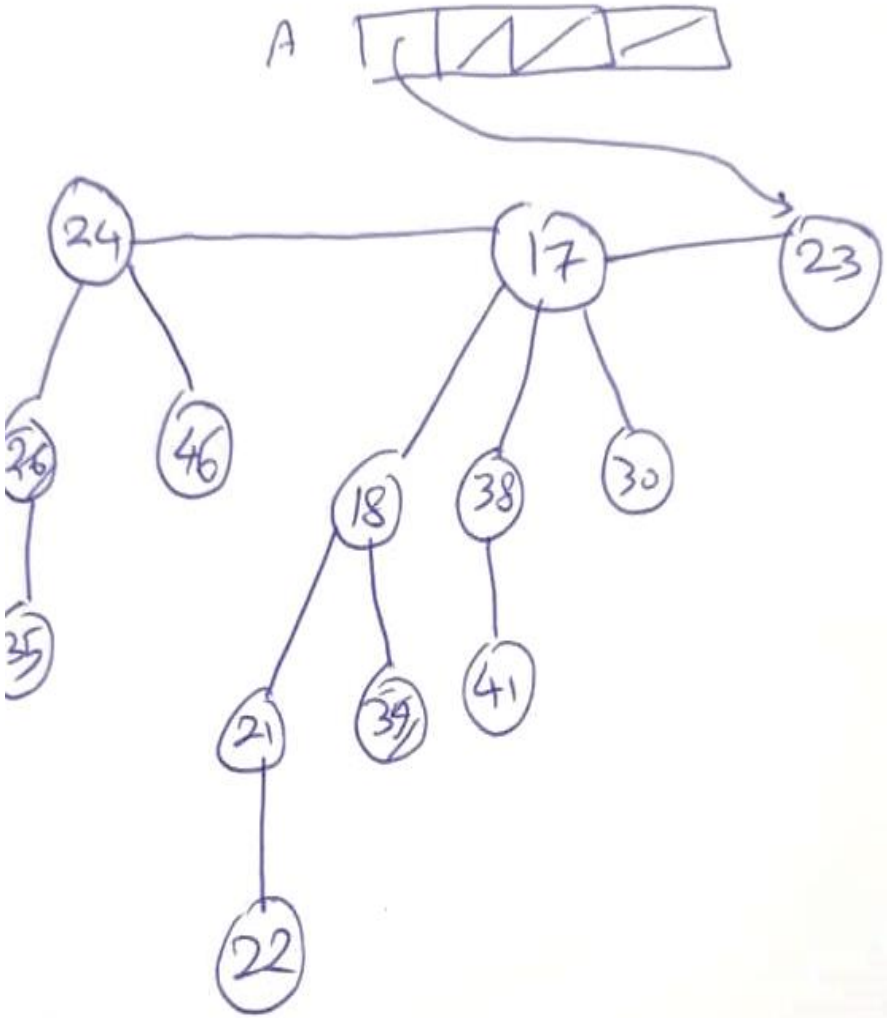


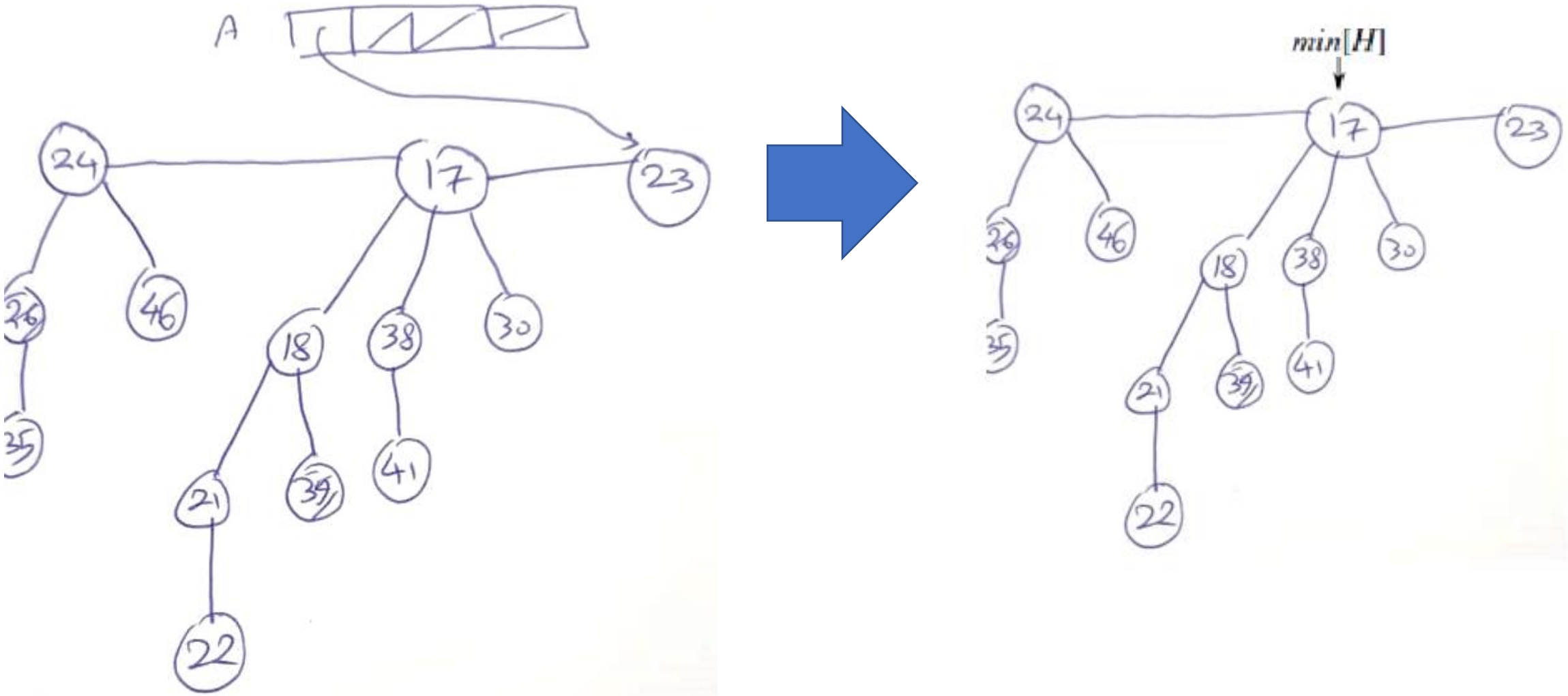






A[2] را با درخت دوجمله ای اخیراً ترکیب شده و هم درجه یک ترکیب می کند.

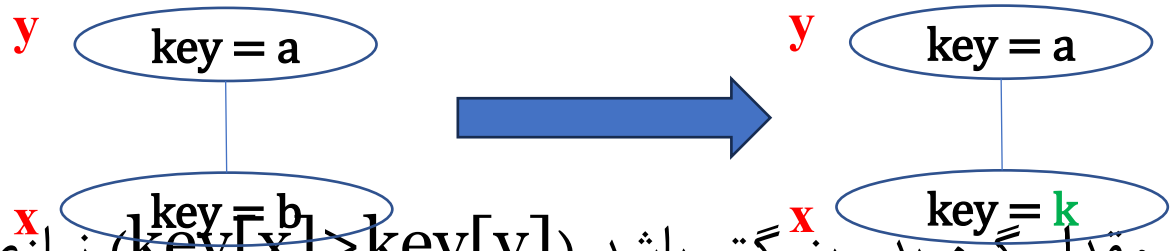



A
24
17
23
26
46
18
38
30
35
21
39
41
22
min[H]
24
17
23
26
46
18
38
30
35
21
39
41
22

عملیات روی هرم های فیبوناچی

۵- کاهش کلید

این عمل، مقدار کلید x (فرضاً گره پدر برابر y باشد) را در هرم فیبوناچی به مقدار جدید k کاهش می دهد (k<x). پس از قرار دادن مقدار k،



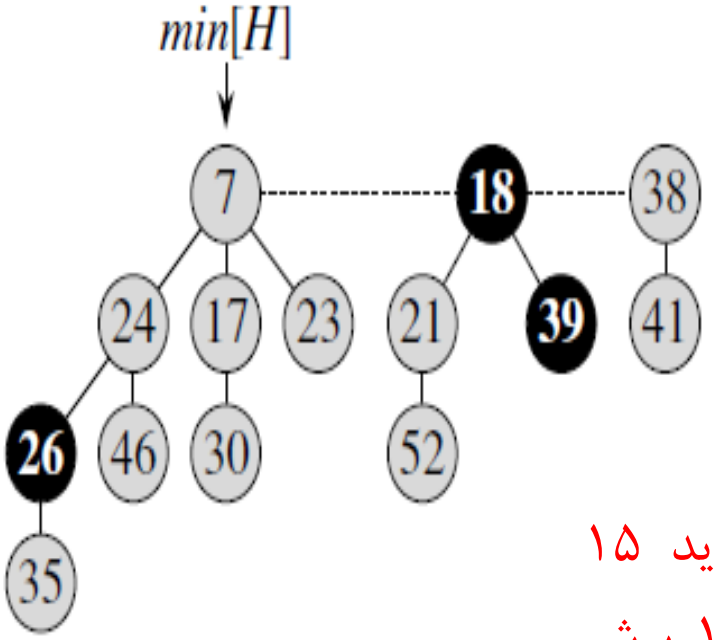
۱- اگر x ریشه باشد و یا مقدار جدیدش (k) از مقدار گره پدر بزرگتر باشد (key[x]>key[y]) نیازی به تغییرات ساختاری نیست.

۲- در غیراینصورت (key[x]<key[y])، گره x برش شده و ارتباط بین پدر و فرزند قطع می شود و گره x بعنوان ریشه درخت جدید و خالی قرار داده می شود. ضمناً mark[x]=False می شود.

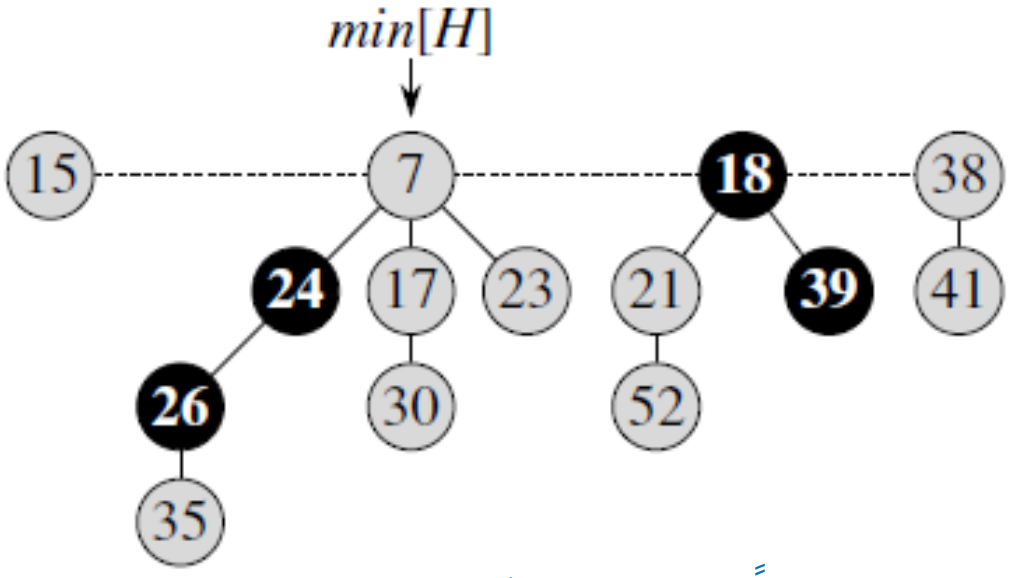
۳- زمانی که پدر x یعنی y به گره دیگری متصل شده است (یعنی فرزند گره دیگری است) آنگاه عمل برش آبشاری را روی y بصورت زیر اجرا می کند:

اگر y ریشه باشد آنگاه الگوریتم به اتمام می رسد در غیر اینصورت اگر y علامتدار نباشد آن را علامتدار کرده و الگوریتم به اتمام می رسد. ولی اگر y علامتدار باشد گره y را برش داده و عمل آبشاری را برای پدر گره y بطور بازگشتی انجام می دهد.

بعنوان مثال: مراحل کاهش مقدار گره ۴۶ را به عدد ۱۵ بنویسید.

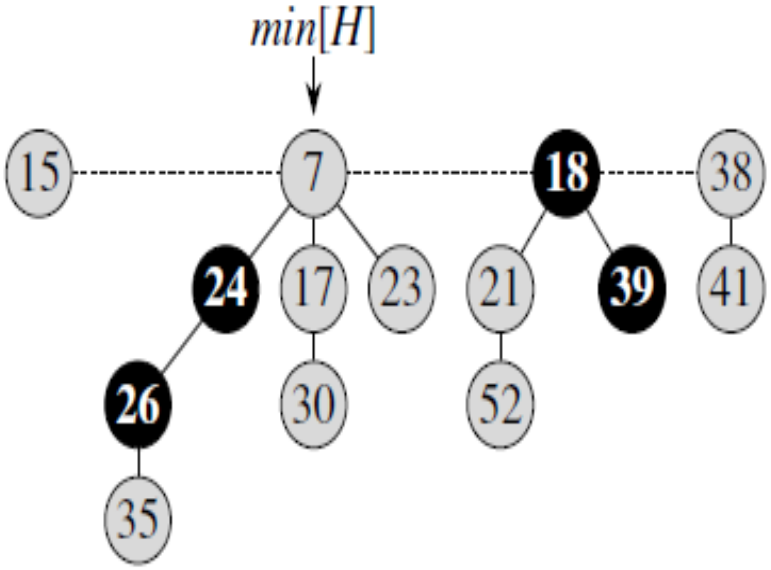


مقدار گره ۴۶ به مقدار جدید ۱۵ تغییر می یابد. سپس گره ۱۵ برش شده و ارتباط بین پدر و فرزند قطع می شود و گره ۱۵ بعنوان ریشه درخت جدید و خالی قرار داده می شود. ضمناً mark[15]=False می شود.

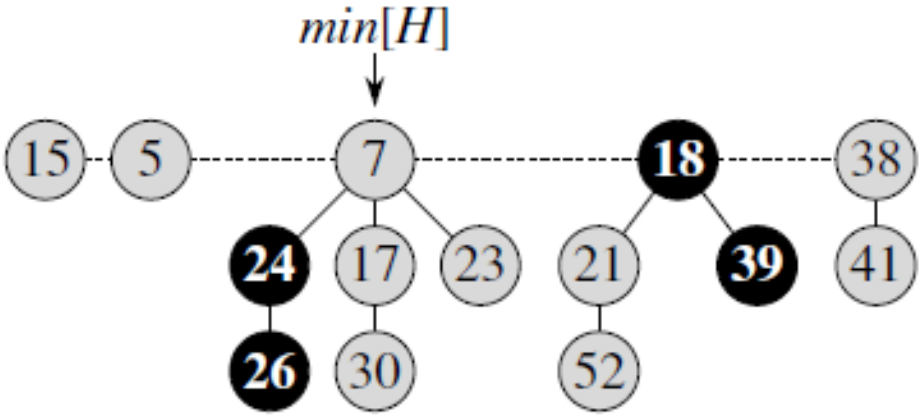


ضمناً، چون پدر گره ۱۵ یعنی ۲۴ به گره دیگری متصل شده است (یعنی فرزند گره دیگری است) آن را علامتدار کرده و الگوریتم به اتمام می رسد.

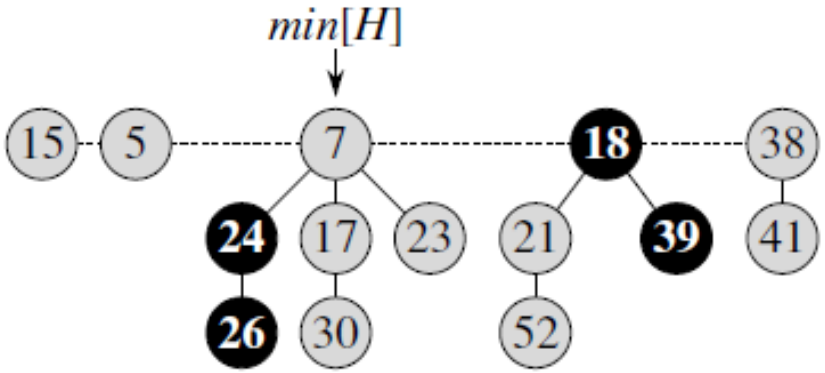
بعنوان مثال: مراحل کاهش مقدار گره ۳۵ را به عدد ۵ بنویسید.



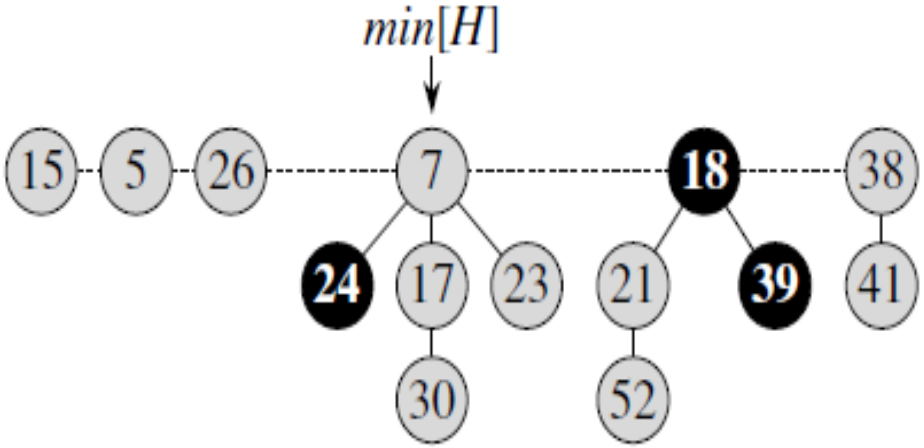
مقدار گره ۳۵ به مقدار جدید ۵ تغییر می یابد. سپس گره ۵ برش شده و ارتباط بین پدر و فرزند قطع می شود و گره ۵ بعنوان ریشه درخت جدید و خالی قرار داده می شود. ضمناً mark[5]=False می شود.



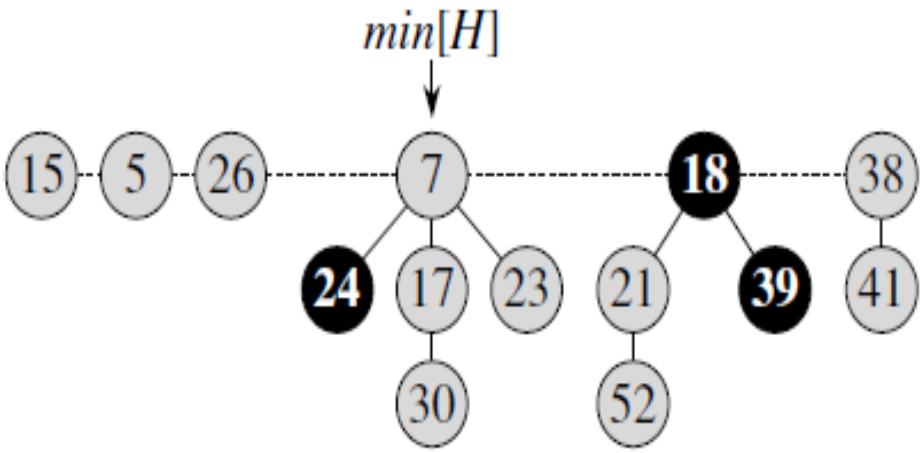
ضمناً، چون پدر گره ۵ یعنی ۲۶ به گره دیگری متصل شده است (یعنی فرزند گره دیگری است) و چون ۲۶ علامتدار بود بطور بازگشتی گره ۲۶ را نیز حذف آبشاری می کند.

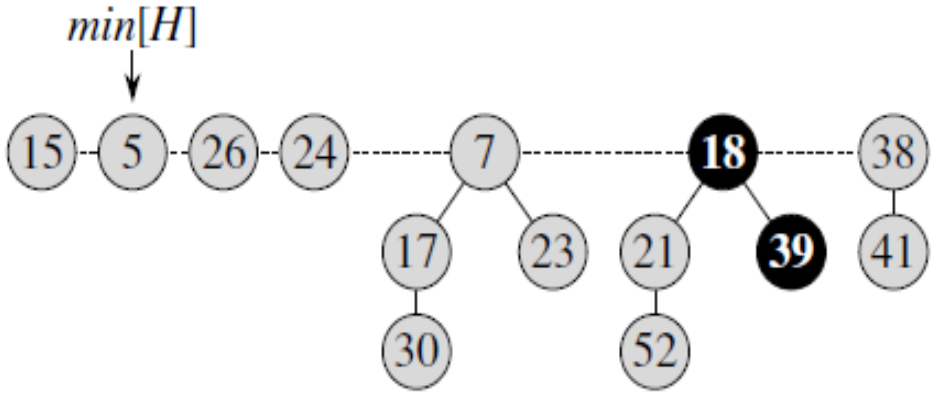


گره ۲۶ برش شده و ارتباط بین پدر و فرزند قطع می شود و گره ۲۶ بعنوان ریشه درخت جدید و خالی قرار داده می شود. ضمناً mark[26]=False می شود.



ضمناً، چون پدر گره ۲۶ یعنی ۲۴ به گره دیگری متصل شده است (یعنی فرزند گره دیگری است) و چون ۲۴ علامتدار بود بطور بازگشتی گره ۲۴ را نیز حذف آبشاری می کند.





گره ۲۴ برش شده و ارتباط بین پدر و فرزند قطع می شود و گره ۲۴ بعنوان ریشه درخت جدید و خالی قرار داده می شود. ضمناً mark[24]=False می شود.

ضمناً، چون پدر گره ۲۴ یعنی ۷ ریشه است آنگاه الگوریتم به اتمام می رسد. و min[H] به ریشه ۵ اشاره می کند.

ثابت می شود که هزینه سرشکن شده برابر هزینه واقعی آن یعنی O(1) یا O(1) است.

۵- حذف یک گره X

برای اینکار، مقدار کلید X را در هرم به مقدار جدید ∞- کاهش می دهد (اجرای الگوریتم کاهش یک کلید) سپس با الگوریتم استخراج کوچکترین مقدار از هرم، آن را حذف می کند.

نکته: زمان اجرای عمل حذف یک گره در هرم برابر است با $O(logn)$

یک نکته در مورد هرم های فیبوناچی

- ✓ k امین عدد فیبوناچی با رابطه بازگشتی زیر تعریف می شود:

$$F_k = \begin{cases} 0 & \text{if } k = 0, \\ 1 & \text{if } k = 1, \\ F_{k-1} + F_{k-2} & \text{if } k \geq 2. \end{cases}$$

- ✓ ثابت می شود که برای همه اعداد صحیح

$$F_{k+2} = 1 + \sum_{i=0}^{k} F_i$$

- ✓

$$F_{k+2} \geq \phi^k \qquad \phi = (1 + \sqrt{5})/2 = 1.61803...$$

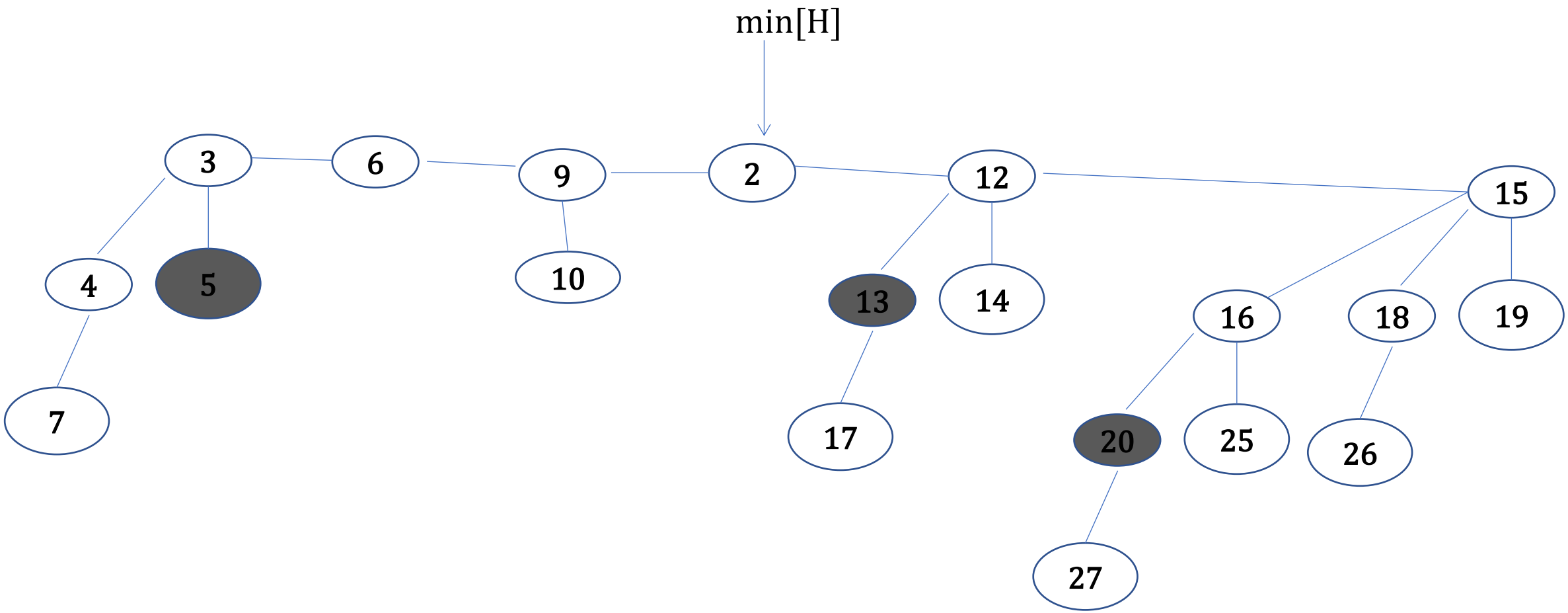
$\text{size}[x]$ تعداد گره های زیردرختی که ریشه اش گره x است را نشان می دهد.

$$\text{size}(x) \geq F_{k+2} \geq \phi^k$$

- ✓ فرض کنید x گره ای از هرم فیبوناچی باشد و همچنین $k = \text{degree}[x]$ داریم:

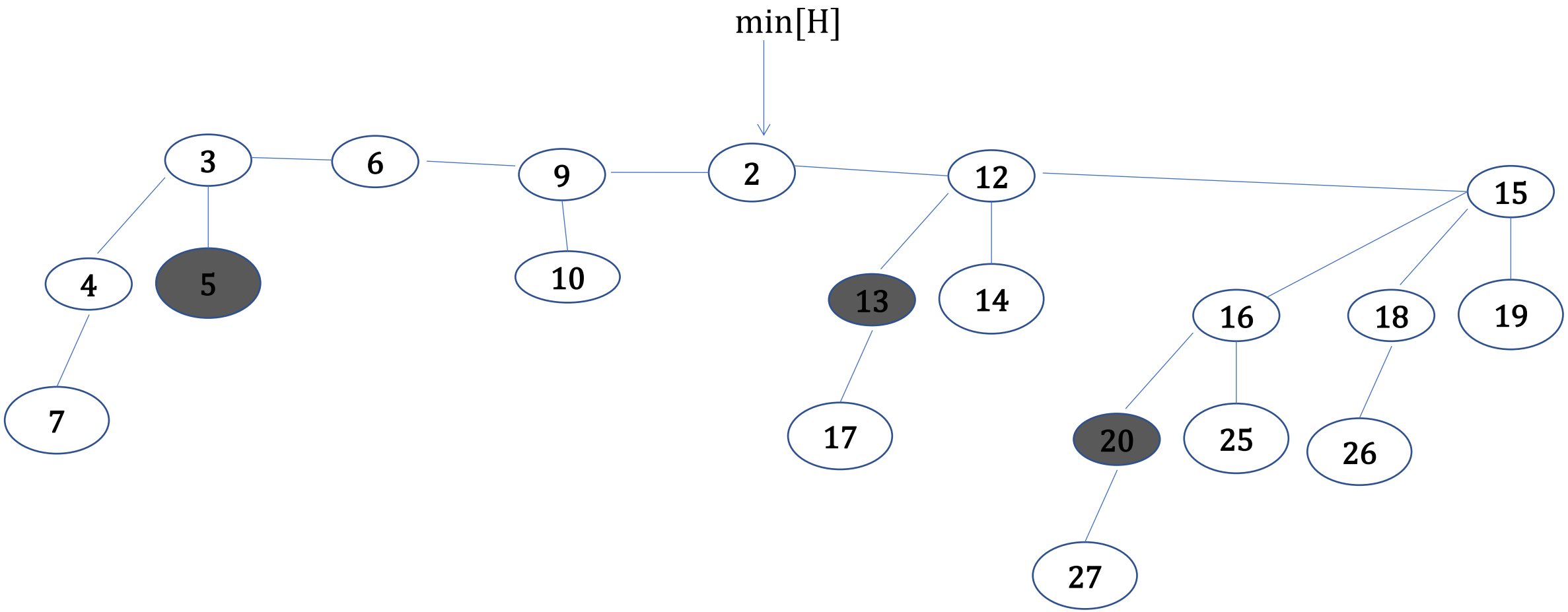
تمرینات سری ۳

۱- مراحل استخراج گره با کمترین مقدار را در هرم زیر بنویسید.



تمرینات سری ۳

۲- مراحل کاهش مقدار گره ۲۷ را به عدد ۱ بنویسید.



پایان فصل ۵