



طراحی الگوریتم ها

فصل ۳: روش حریصانه

لیلا ناموری تازه کند

نیمسال دوم سال تحصیلی ۱۴۰۳-۱۴۰۴

○ مفهوم روش حریصانه (Greedy)

○ مساله ۱: درخت پوشای کمینه (Minimum Spanning Tree) یا (MST)

○ مساله ۲: الگوریتم دایکسترا برای کوتاهترین مسیر تک منبع (Dijkstra)

○ مساله ۳: مساله کوله پشتی

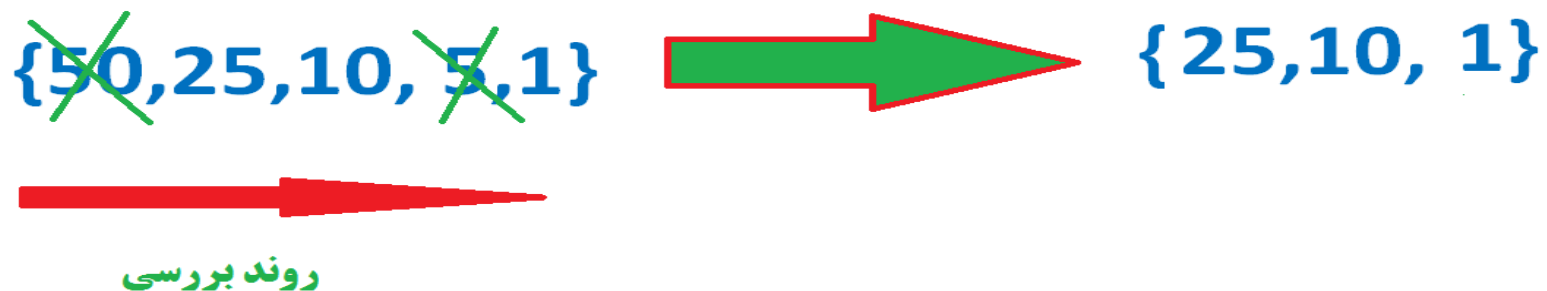
○ مساله ۴: مساله زمانبندی

✓ مسائلی که با روش حریصانه قابل حل هستند دارای خصوصیات زیر می باشند:

1. مساله، بهینه سازی (Optimization) است (کمترین، بهترین، بیشترین و...).
2. برای حل بهینه مساله، باید زیرمساله های آن را نیز بصورت بهینه حل کرد.
3. انتخاب حریصانه (Greedy Choice) در این گونه مسائل بهترین انتخاب است و عوض نمی شود (تفاوت با روش برنامه نویسی پویا در این ویژگی است).

✓ مثال: خرد کردن پول $n=36$ ریال را بین سکه های $\{1,5,10,25,50\}$ با حداقل تعداد سکه ها (تکرار سکه ها مجاز است)

جواب: از سکه ۵۰ ریالی شروع می کنیم و یکی یکی بررسی می کنیم:



نکته: الگوریتم حریصانه همواره به حل بهینه ختم نمی شود.

مثال: $n=16$ و مجموعه سکه ها عبارتند از $\{50, 25, 12, 10, 5, 1\}$
جواب:

- $\text{روش حریصانه} = \{12, 1, 1, 1, 1\}$
- $\text{روش بهینه} = \{10, 5, 1\}$

✓ بطور خلاصه: الگوریتم حریصانه بصورت زیر کار می کند:

این الگوریتم، کار را با یک مجموعه تهی آغاز کرده و به ترتیب، عناصری به مجموعه جواب اضافه می کند که این مجموعه، راه حلی برای نمونه ای از مساله را نشان می دهد. در هر مرحله، عملیات زیر را تکرار می کند:

- **روال انتخاب (select)**، عنصر بعدی را که باید به مجموعه اضافه شود، انتخاب می کند. انتخاب براساس یک ملاک حریصانه انجام می شود که یک شرط بهینه را در همان برهه برآورده می سازد.

- **بررسی امکان سنجی (feasible)**، تعیین می کند که آیا مجموعه جدید برای رسیدن به حل نمونه عملی است یا خیر.

- **بررسی راه حل (solution)**، بررسی می کند که آیا مجموعه جدید، راه حلی برای نمونه مساله می باشد یا خیر.

```

set Greedy (set C)  // C is a set of candidates
{
    set S=NULL;    //S is a set of solution
    While (! solution(S) && !empty(C))
    {
        x = select(C);
        C = C - { x };
        if (feasible (S  $\cup$  { x }))
            S= S  $\cup$  { x };
    }
    if (solution(S))
        return S;
    else
        return NULL;
}

```

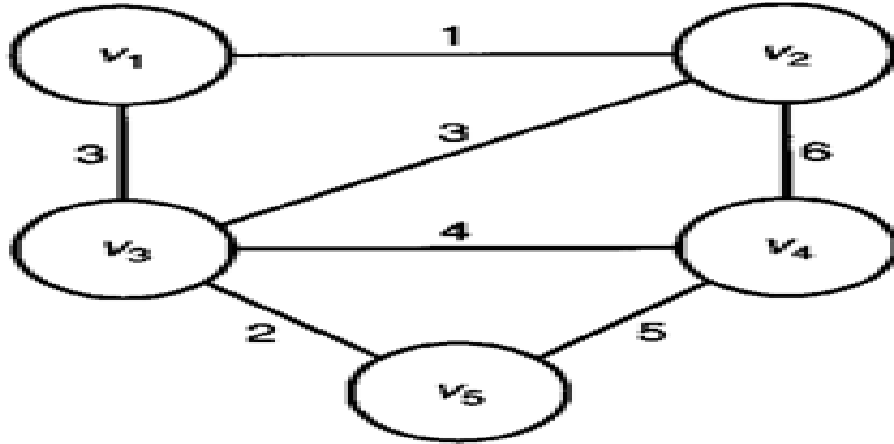
✓ **تعریف:** درخت پوشا برای گراف G با n راس،

1. یک زیرگراف متصل است که شامل همه رئوس در G می باشد.

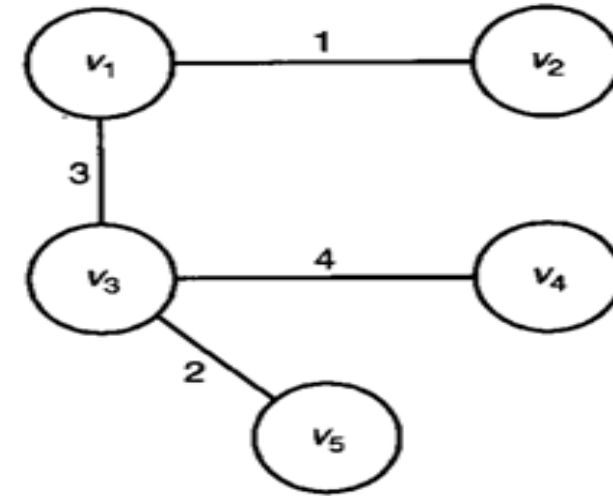
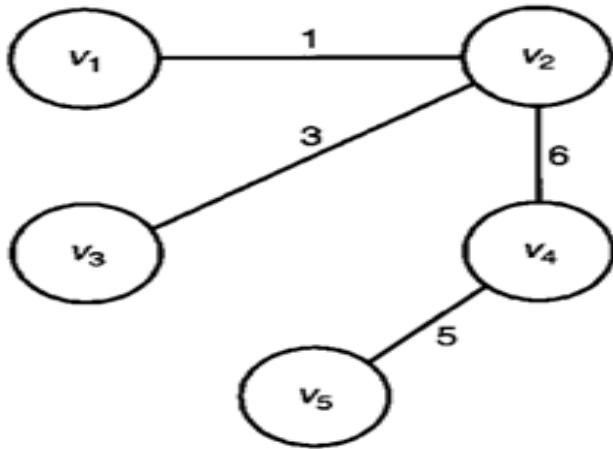
2. یک درخت می باشد یعنی دارای دور نیست و تعداد یال های آن برابر $n-1$ است.

✓ درخت پوشا با کمترین وزن را یک MST می گویند.

✓ **مثال:** چند درخت پوشا در گراف زیر را پیدا کنید؟ ضمناً MST را نیز مشخص کنید؟



جواب: تعداد رئوس برابر $n=5$ است بنابراین درخت پوشا باید ۵ راس و ۴ یال داشته باشد یعنی درخت حاصله دور نداشته باشد. شکل زیر دو نمونه را نشان می دهد.



$1+3+4+2=10$ = مجموع وزن

$1+3+5+6=15$ = مجموع وزن

بنابراین، این درخت، یک MST (درخت پوشای کمینه) برای گراف داده شده محسوب می شود.

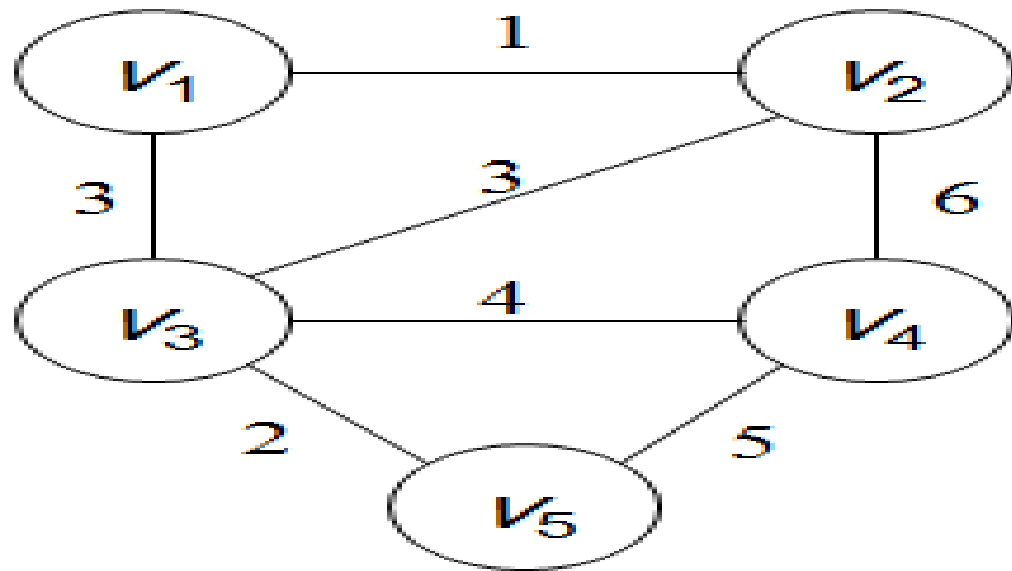
✓ فرض کنیم گراف $G=(V,E)$ را داریم مراحل الگوریتم پریم بصورت زیر می باشد:

➤ مرحله ۱: این الگوریتم با مجموعه ای تهی از یالها بنام F و زیر مجموعه ای از رئوس Y آغاز می شود و در ابتدا داریم : $Y=\{v_1\}$, $F=\{\}$

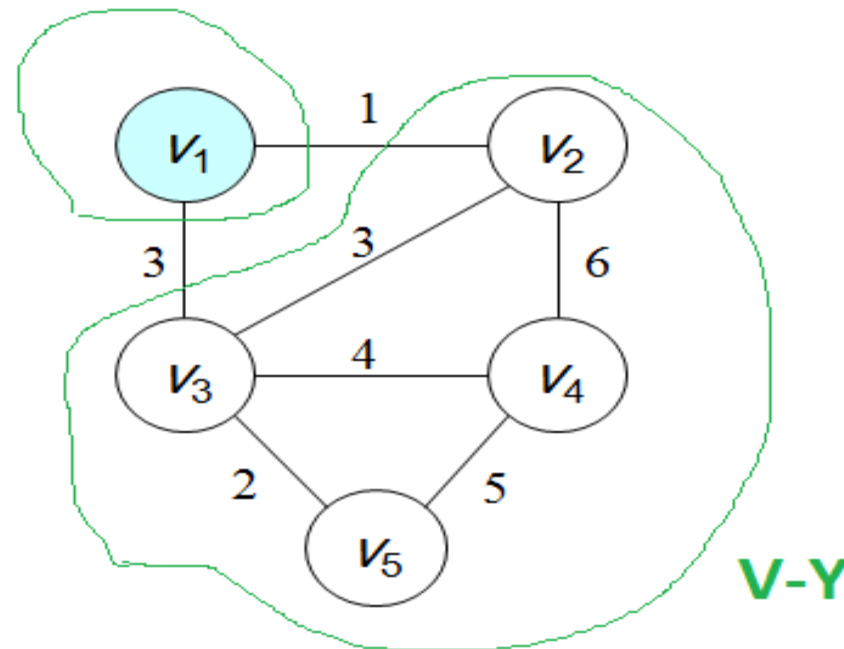
➤ مرحله ۲: رأسی از مجموعه $V-Y$ را که نزدیکترین راس به رئوس مجموعه Y است را انتخاب و آنرا به مجموعه Y اضافه می کنیم همچنین یال مربوطه را نیز به F اضافه می کنیم به این شرط که دور ایجاد نشود.

➤ مراحل ۱ و ۲ را تا موقعی ادامه می دهیم که $V=Y$ بشود یعنی تمامی رئوس انتخاب شوند.

✓ مثال: درخت MST را با روش پریم در گراف زیر پیدا کنید؟



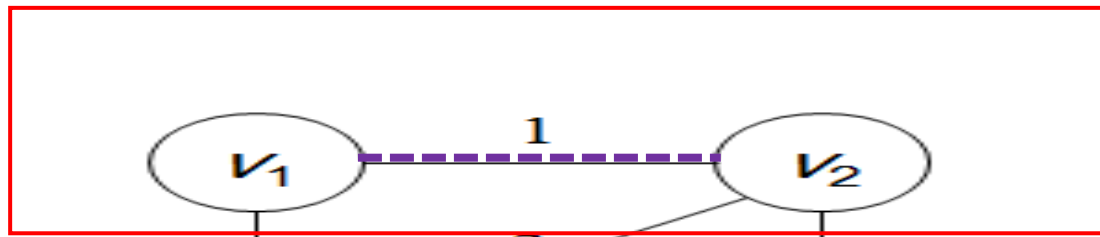
Y



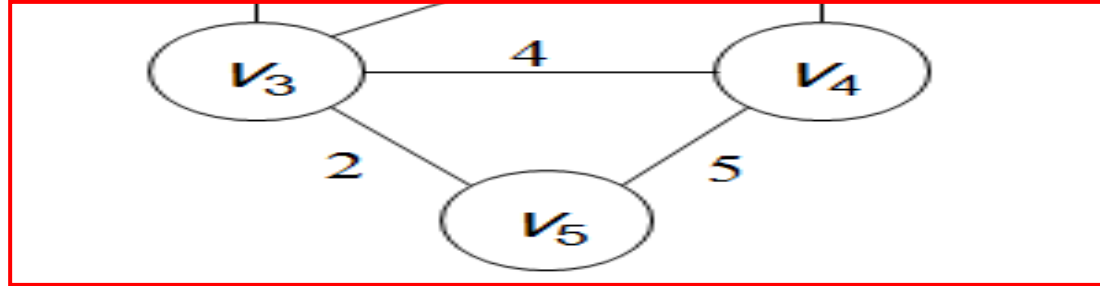
✓ جواب:

مرحله ۱:

Y

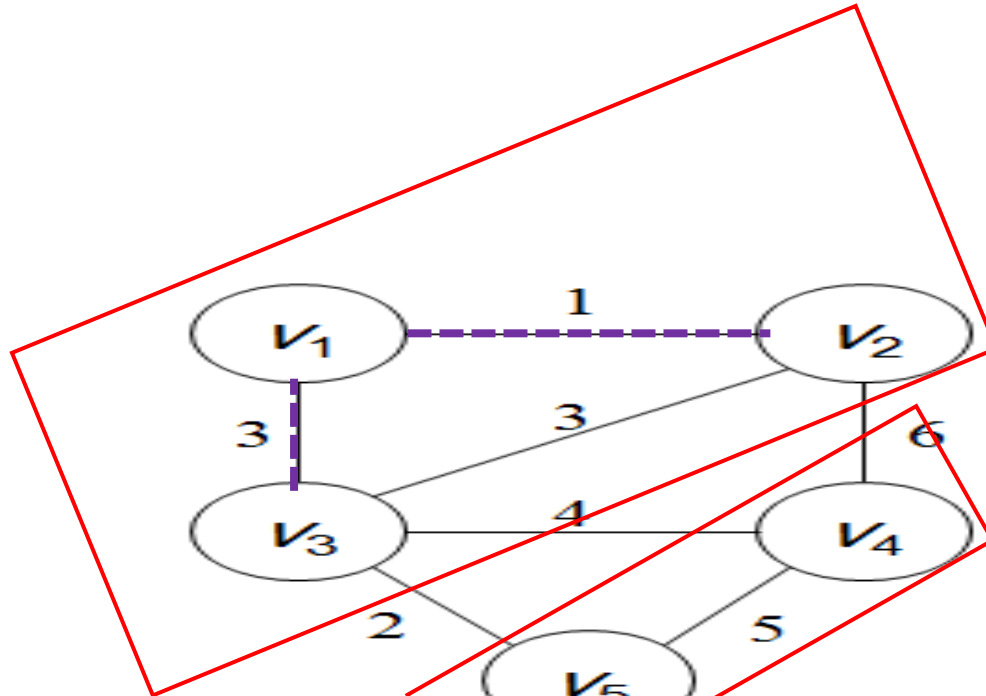


V-Y



✓ مرحله ۲: انتخاب راس V_2

Y

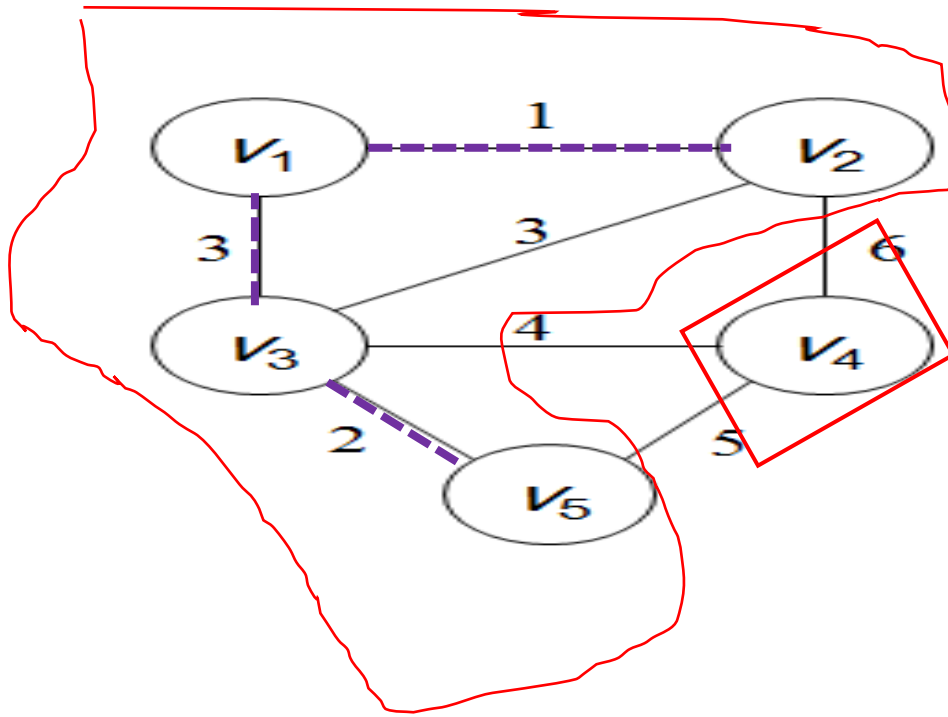


V-Y

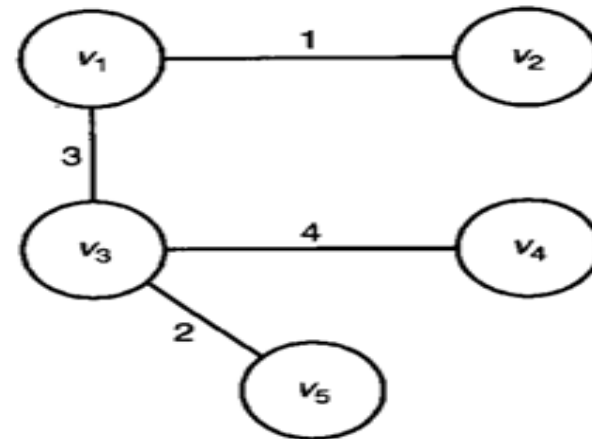
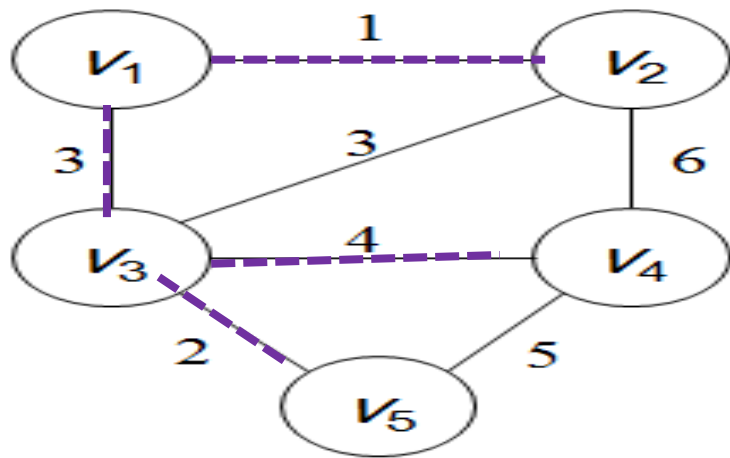
✓ مرحله ۳: انتخاب راس V_3

✓ مرحله ۴: انتخاب راس V_5

Y



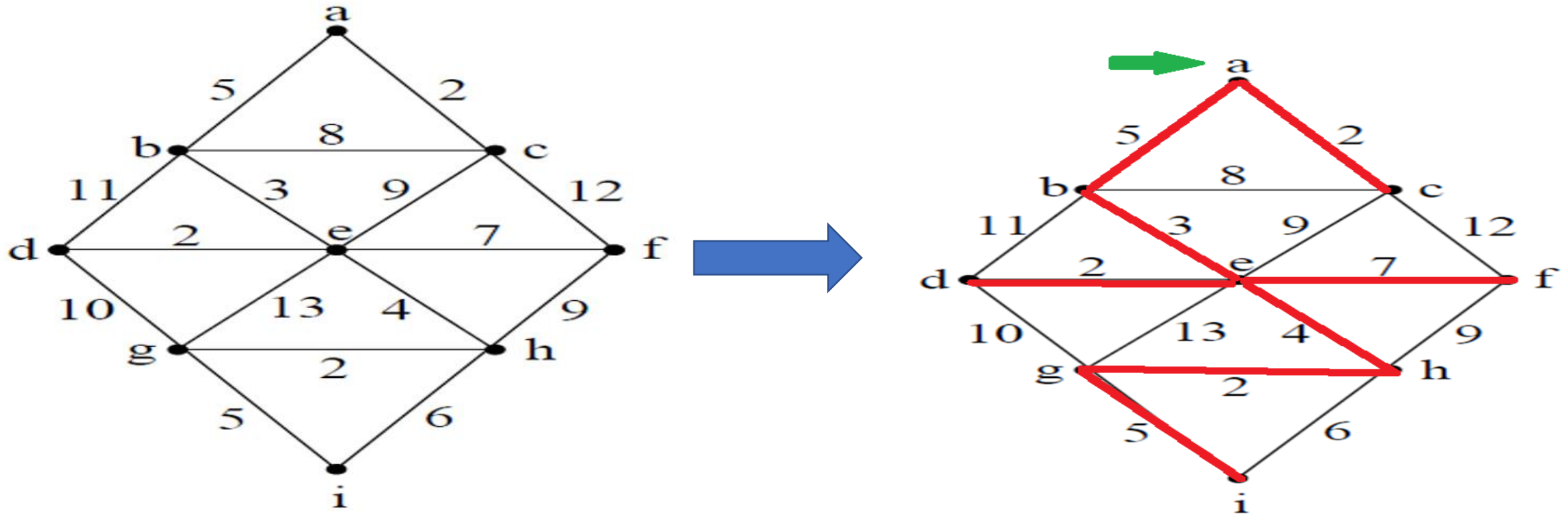
V-Y



✓ مرحله ۵: انت

$$\text{مجموع وزن} = 1 + 3 + 4 + 2 = 10$$

✓ مثال: درخت MST را با روش پریم در گراف زیر پیدا کنید؟



$$\text{مجموع وزن} = 5 + 2 + 3 + 2 + 4 + 2 + 5 + 7 = 30$$

✓ پیچیدگی زمانی: فرضاً n تعداد رئوس باشد.

$$T(n) = (n-1) * (n-1) = \theta(n^2)$$

تعداد تکرار : چون
همه رئوس باید به
مجموعه Y اضافه شوند

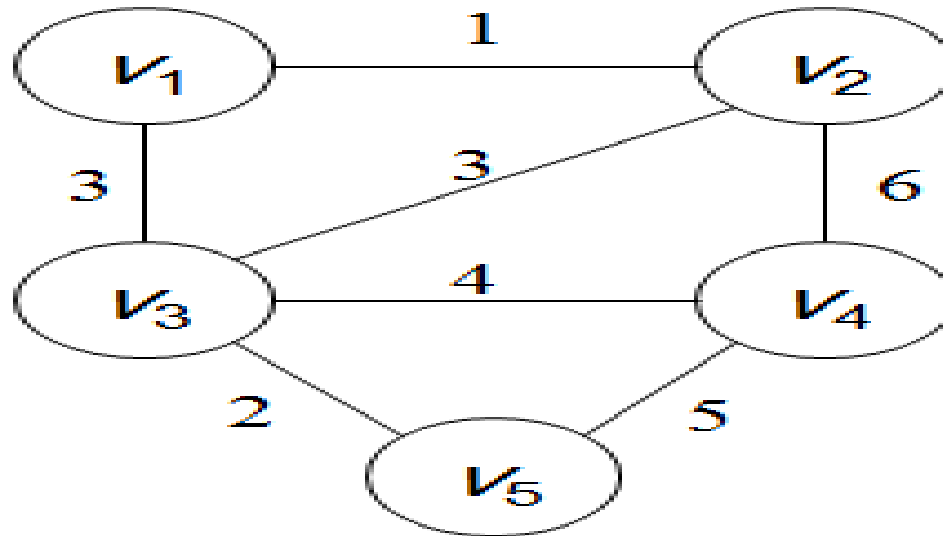
انتخاب کوتاهترین راس

✓ در این الگوریتم، ابتدا کلیه یال ها را بر اساس وزن آنها مرتب کرده و سپس از یال با کمترین وزن شروع کرده و بعد یالهای بعدی را در نظر می گیریم تا موقعی که دور پیش نیاید و تعداد یالهای انتخابی برابر $n-1$ شود.

✓ پیچیدگی زمانی: فرضا n تعداد رئوس و m تعداد یالها باشد.

$$T(m,n) = \theta(m \log m)$$

✓ مثال: درخت MST را با روش کروسکال در گراف زیر پیدا کنید؟



✓ مرحله ۱: مرتب سازی یالها براساس وزن آنها

(v_1, v_2) , (v_3, v_5) , (v_1, v_3) , (v_2, v_3) , (v_3, v_4) , (v_4, v_5) , (v_2, v_4)

✓ مرحله ۲: انتخاب یالها از کمترین وزن به بیشترین وزن به شرط اینکه دور ایجاد نشود و تعداد یالهای انتخابی هم برابر $n-1$ باشد.

✓ مقایسه الگوریتم پریم و کروسکال: فرضا n تعداد رئوس و m تعداد یالها باشد.

□ الگوریتم پریم: $T(m,n) = \theta(n^2)$

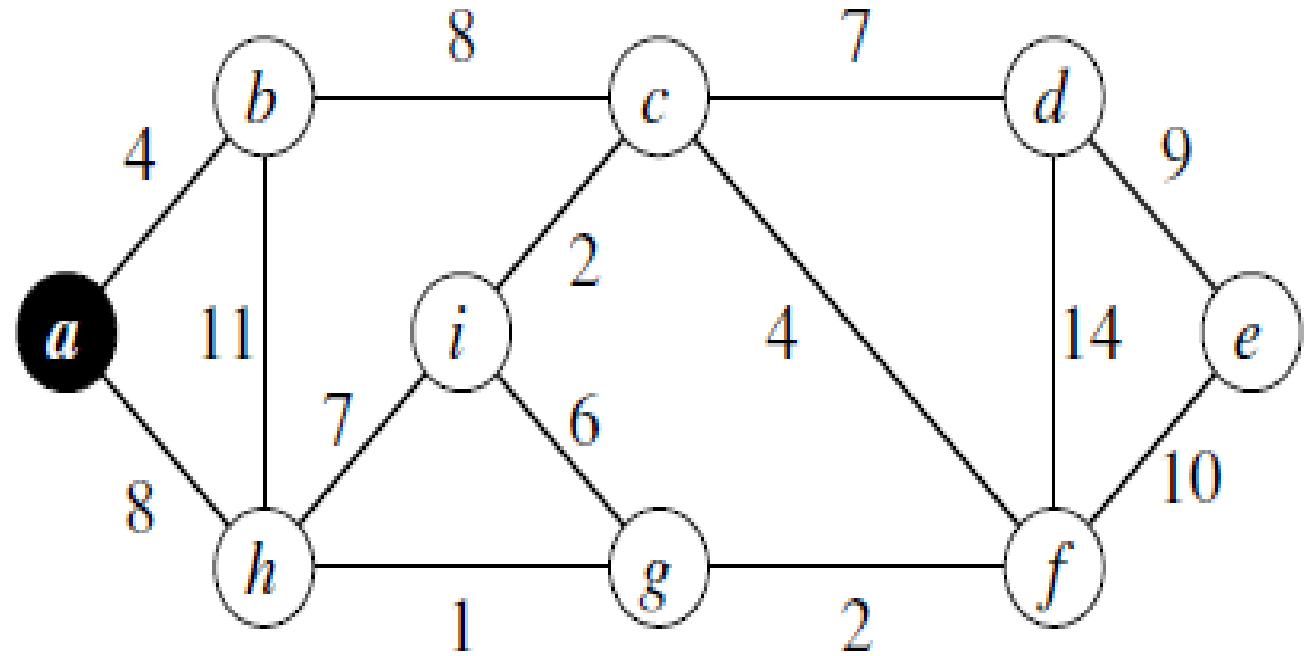
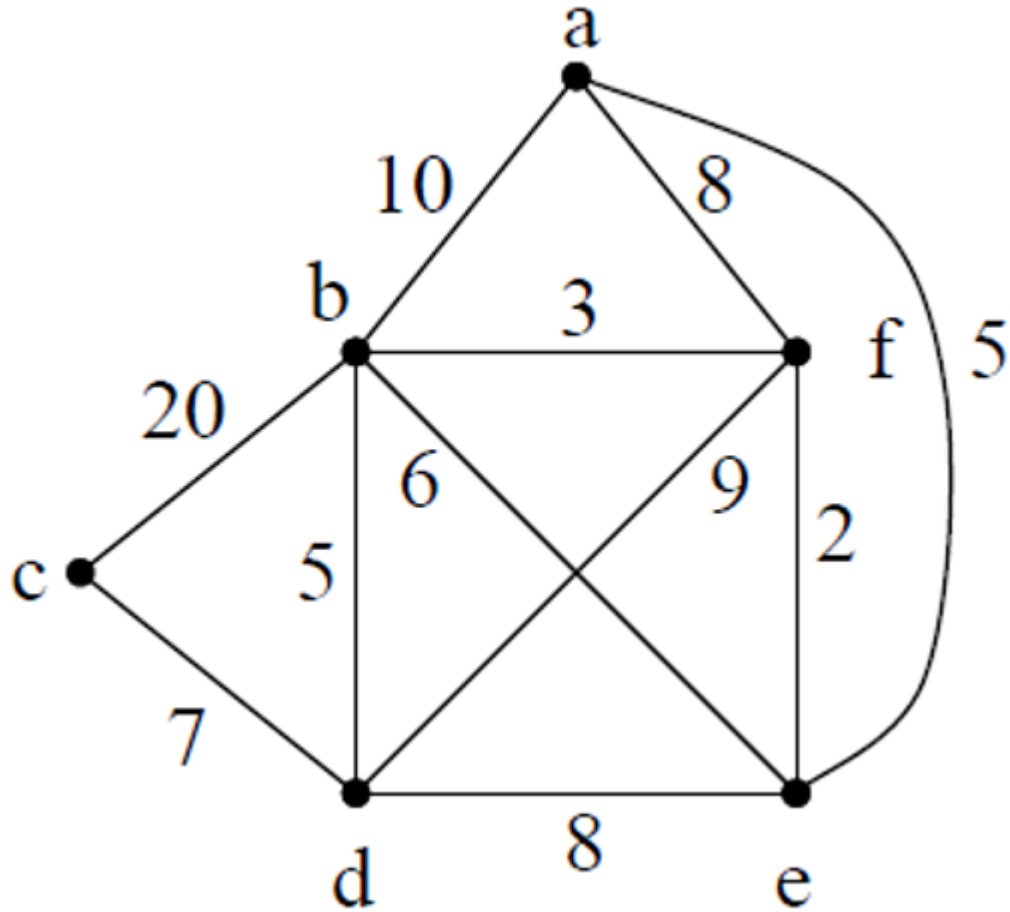
□ الگوریتم کروسکال: $T(m,n) = \theta(m \log m)$

✓ اگر گراف بسیار متصل (تقریبا کامل) باشد یعنی $m \cong n^2$ باشد آنگاه الگوریتم پریم بهتر است.

✓ اگر گراف بسیار خلوت باشد یعنی $m \cong n$ باشد آنگاه الگوریتم کروسکال بهتر است.

تمرینات سری ۱

۱- درخت MST را با روش پریم و کروسکال در گرافهای زیر پیدا کنید؟



مساله ۲: الگوریتم دایکسترا برای کوتاهترین مسیر تک منبع (Dijkstra)

✓ **قبلا،** الگوریتم فلویید با پیچیدگی $\theta(n^3)$ را برای تعیین کوتاهترین مسیر مابین تمام رئوس در یک گراف وزن دار و جهت دار ارائه کرده بودیم که از برنامه نویسی پویا استفاده می کرد.

✓ در این جا، از روش حریصانه استفاده می کنیم برای ارائه یک الگوریتم از درجه $\theta(n^2)$ جهت محاسبه کوتاهترین مسیر از یک راس به بقیه رئوس دیگر.

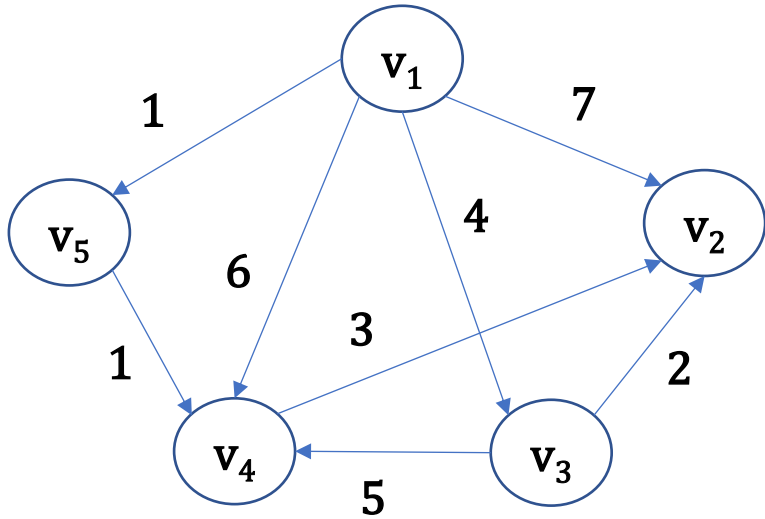
✓ این الگوریتم مشابه الگوریتم پریم برای مساله MST است. ابتدا راس v_1 را در مجموعه Y قرار می دهیم که قرار است کوتاهترین مسیر از این راس را به بقیه رئوس تعیین کنیم. پس $Y = \{v_1\}$. ضمنا در مجموعه F نیز، یالها را نگهداری می کنیم و در ابتدا داریم $F = \{\}$

✓ راس v از مجموعه $V - Y$ را طوری انتخاب می کنیم که کوتاهترین مسیر از v_1 به v با استفاده از رئوس موجود در Y باشد. سپس راس v را به Y و یال $\langle v_1, v \rangle$ را به مجموعه F اضافه می کنیم.

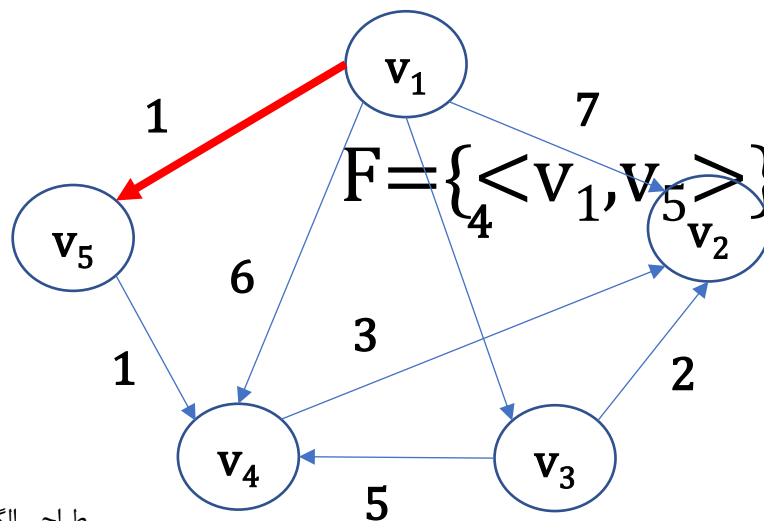
✓ مرحله قبلی را تا موقعی تکرار می کنیم که $Y = V$ بشود. یعنی تمام رئوس انتخاب شده باشند.

✓ **پیچیدگی الگوریتم دایکسترا:** چون باید $n-1$ راس به مجموعه Y اضافه شوند و در هر مرحله، محاسبه کوتاهترین مسیر دارای پیچیدگی $\theta(n)$ می باشد بنابراین پیچیدگی این الگوریتم برابر $\theta(n^2)$ خواهد بود.

✓ **مثال:** با استفاده از الگوریتم دایکسترا، کوتاهترین مسیر از V_1 به بقیه رئوس را در گراف زیر پیدا کنید؟

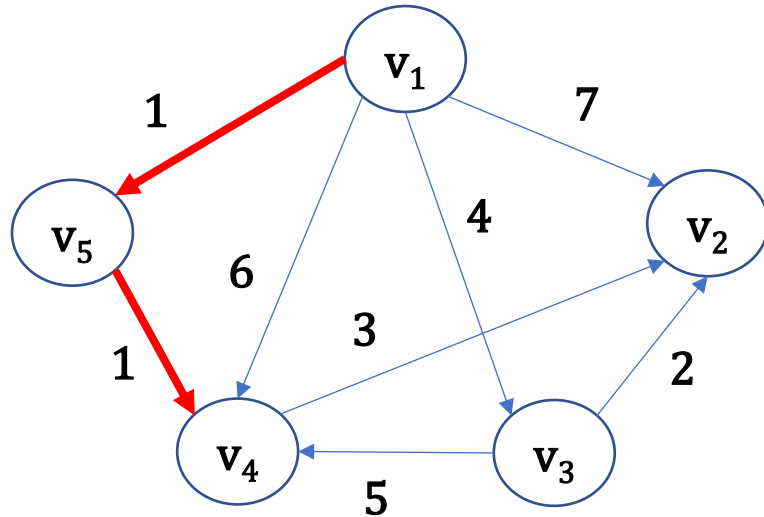


جواب: در ابتدا داریم: $Y = \{v_1\}$ و $F = \{\}$

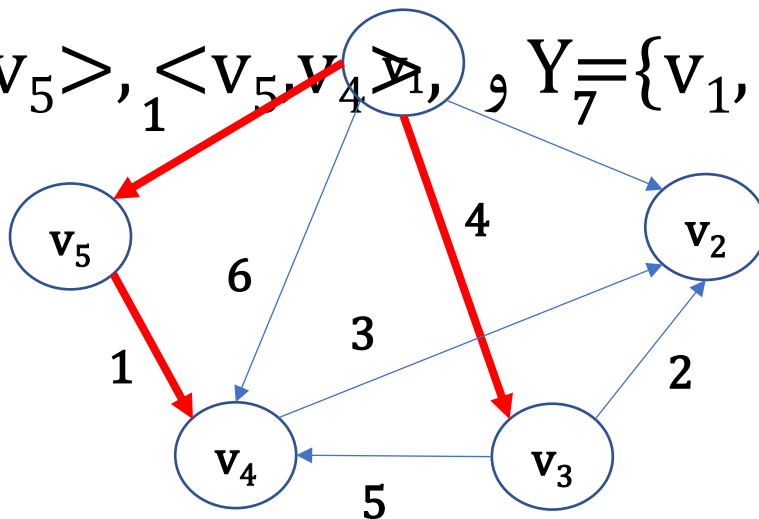


مرحله ۱: انتخاب راس v_5 در نتیجه $Y = \{v_1, v_5\}$ و $F = \{ \langle v_1, v_5 \rangle \}$

مرحله ۲: انتخاب راس v_4 در نتیجه $Y = \{v_1, v_5, v_4\}$ و $F = \{<v_1, v_5>, <v_5, v_4>\}$

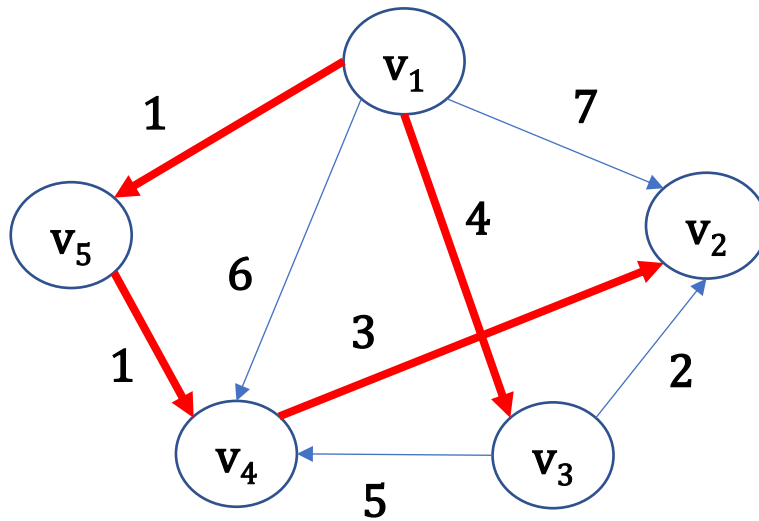


مرحله ۳: انتخاب راس v_3 در نتیجه $Y = \{v_1, v_5, v_4, v_3\}$ و $F = \{<v_1, v_5>, <v_5, v_4>, <v_4, v_3>\}$ و $<v_1, v_3>\}$



مرحله ۴: انتخاب راس v_2 در نتیجه $Y = \{v_1, v_5, v_4, v_3, v_2\}$ و

$$F = \{ \langle v_1, v_5 \rangle, \langle v_5, v_4 \rangle, \langle v_1, v_3 \rangle, \langle v_4, v_2 \rangle \}$$



✓ در گراف حاصله، کوتاهترین مسیرها از راس v_1 به همه رئوس دیگر پیدا شده اند.

✓ خلاصه مساله:

- یک کوله پشتی با ظرفیت W
 - n نوع کالا $S = \{item_1, \dots, item_n\}$
 - وزن کالای i ام برابر w_i
 - ارزش کالای i ام برابر p_i
- فرضا همه مقادیر صحیح هستند.

هدف این است که کوله پشتی را تا حد ممکن پر کنیم که مجموع ارزش کالاها ماکزیمم شود.

✓ راه حل حریصانه: کالاها را به ترتیب کاهش ارزش آنها بازای واحد وزن مرتب می کنیم و آنها را به ترتیب انتخاب می کنیم. هر قطعه در صورتی در کوله پشتی گذاشته می شود که وزن کل آن از W فراتر نرود.

✓ مثال: ارزش بهینه را برای مساله کوله پشتی زیر با روش حریصانه محاسبه کنید.

$$S=\{\text{item}_1, \text{item}_2, \text{item}_3\} \quad W=30 \quad n=3$$

$$w=\{5,10,20\}$$

$$p=\{50\$,60\$,140\$\}$$

جواب: ارزش قطعات بازای واحد وزن آنها عبارتند از:

$$\text{item}_1=50\$/5=10\$ \quad \text{item}_2=60\$/10=6\$ \quad \text{item}_3=140\$/20=7\$$$

ترتیب کالاها با کاهش ارزش بازای واحد وزن عبارت است از:

$$\text{item}_1, \text{item}_3, \text{item}_2$$

پس طبق الگوریتم، کالاهای item_1 و item_3 انتخاب می شوند که منفعت کل برابر $190\$$ می شود.

نکته: اگر از برنامه نویسی پویا استفاده بکنیم کالاهای item_2 و item_3 انتخاب می شوند که منفعت کل برابر $200\$$ می شود.

✓ از روش حریصانه ارائه شده می توان برای حل **مساله کوله پشتی کسری** نیز استفاده کرد.

✓ **مثال:** ارزش بهینه را برای مساله کوله پشتی کسری زیر با روش حریصانه محاسبه کنید.

$$S=\{item_1, item_2, item_3\} \quad W=30 \quad n=3$$

$$w=\{5,10,20\}$$

$$p=\{50\$,60\$,140\$\}$$

جواب: ارزش قطعات بازای واحد وزن آنها عبارتند از:

$$item_1=50\$/5=10\$ \quad item_2=60\$/10=6\$ \quad item_3=140\$/20=7\$$$

ترتیب کالاها با کاهش ارزش بازای واحد وزن عبارت است از:

$$item_1, item_3, item_2$$

پس طبق الگوریتم، کالاهای $item_1$ و $item_3$ بطور کامل و نصف کالای $item_2$ انتخاب می شوند که منفعت کل برابر $50\$+140\$+0.5*60\$=220\$$ خواهد بود.

نکته: روش کسری هیچ ظرفیتی را هدر نمی دهد و همواره حل بهینه را بدست می دهد.

✓ **توضیح:** تعدادی کار در سیستم موجود است. هدف این است که این کارها به ترتیبی انجام شوند که زمان کل در سیستم (زمان انتظار+زمان اجرا) کمینه شود.

✓ **یک راه حل ساده و غیرهوشمندانه:** همه زمانبندیهای ممکن را در نظر گرفته سپس آن را که کمینه است انتخاب می کنیم. زمان چنین الگوریتمی برابر $n!$ است که n تعداد کارها می باشد.

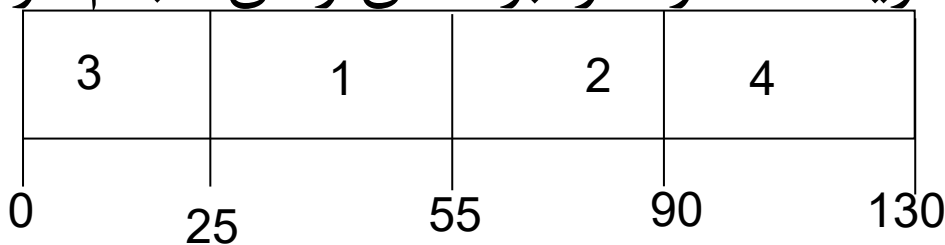
✓ **الگوریتم حریرانه:** ابتدا کارها را براساس مدت زمان انجام مرتب کن سپس آنها را به ترتیب انجام بده.

✓ پیچیدگی این روش برابر است با $\theta(n \log n)$

✓ **مثال:** ترتیب انجام کارها به چه صورت باشد تا زمان کل در سیستم کمینه شود؟ ضمناً زمان کل را محاسبه کنید.

کار	زمان انجام
۱	۳۰
۲	۳۵
۳	۲۵
۴	۴۰

جواب: طبق الگوریتم حریصانه، کارها را براساس زمان انجام مرتب کرده و به ترتیب انجام می شوند:



کار	زمان انتظار	زمان انجام	زمان انتظار+زمان انجام
1	25	30	55
2	55	35	90
3	0	25	25
4	90	40	130

مجموع کل برابر است با $۱۳۰+۲۵+۹۰+۵۵ = ۳۰۰$ واحد زمانی

✓ **زمانبندی با مهلت معین:** در این نوع زمانبندی، هر کاری در **یک واحد زمانی** به اتمام می رسد و دارای مهلت و بهره مشخصی برای انجام می باشد. اگر کاری پیش از اتمام مهلت مشخص شده انجام شود بهره بدست می آید.

✓ هدف این است که زمانبندی به نحوی انجام شود که بهره بیشینه بدست آید. لازم نیست که حتما همه کارها انجام شوند.

✓ هر زمانبندی که در آن، یک کار پس از مهلت مقرر آن برنامه ریزی شده باشد آن زمانبندی را غیرممکن می نامیم.

✓ **مثال:** فرضا چهار کار با مهلت و بهره مشخص شده داریم

بهره	مهلت	کار
۳۰	۲	۱
۳۵	۱	۲
۲۵	۲	۳
۴۰	۱	۴

(فرض کرده ایم که هر کار فقط در یک واحد زمانی انجام شود ضمنا ساعت شروع سیستم صفر تنظیم شده است):

زمانبندی [1,2] غیرممکن است. چون کار ۲ باید در زمان ۱ تمام شود.

زمانبندی [1,3] ممکن است و دارای بهره ۵۵ می باشد.

زمانبندی [2,1,3] غیرممکن است. چون کار 3 باید در زمان 2 تمام شود.

در اسلاید بعدی تمام زمانبندی های ممکن ذکر شده اند.

بهره	مهلت	کار
۳۰	۲	۱
۳۵	۱	۲
۲۵	۲	۳
۴۰	۱	۴

تمام زمانبندی های ممکن و بهره ها بصورت زیر می باشند:

زمانبندی	بهره کل
[1,3]	$30+25=55$
[2,1]	$35+30=65$
[2,3]	$35+25=60$
[3,1]	$25+30=55$
[4,1]	$40+30=70$
[4,3]	$40+25=65$

در نتیجه، زمانبندی [4,1] بهینه است.

✓ **روش حریصانه:** ابتدا کارها را براساس بهره آنها بصورت نزولی مرتب کرده سپس هر کدام از کارها را به ترتیب و در صورت امکان به زمانبندی اضافه می کنیم.

بهره	مهلت	کار
۴۰	۳	۱
۳۵	۱	۲
۳۰	۱	۳
۲۵	۳	۴
۲۰	۲	۵

✓ **مثال:** فرضاً ۵ کار با مهلت و بهره مشخص شده داریم:

قبل از شماره گذاری کارها، آنها را براساس بهره مرتب نزولی کرده ایم.

جواب: در ابتدا، زمانبندی $S=[]$ را در نظر می گیریم.

۱- بررسی کار ۱: $S=[1]$

۲- بررسی کار ۲: $S=[2,1]$

۳- بررسی کار ۳: این کار نمی تواند در مهلت مقرر انجام شود.

۴- بررسی کار ۴: $S=[2,1,4]$

۵- بررسی کار ۵: این کار نمی تواند در مهلت مقرر انجام شود.

مقدار نهایی برابر $S=[2,1,4]$ می باشد که دارای بهره $35+40+25=100$ می باشد.

پیچیدگی این روش:

$$T(n) = n \log n + n.n = \theta(n^2)$$

n بار کاری انتخاب شده و زمانبندی بررسی می شود. زمان مرتب سازی

✓ **کار در کلاس:** فرضاً ۵ کار با مهلت و بهره مشخص شده داریم:

قبل از شماره گذاری کارها، آنها را براساس بهره مرتب نزولی کرده ایم.

جواب: در ابتدا، زمانبندی $S=[]$ را در نظر می گیریم.

۱- بررسی کار ۱: $S=[1]$

۲- بررسی کار ۲: $S=[2,1]$

۳- بررسی کار ۳: $S=[2,1,3]$

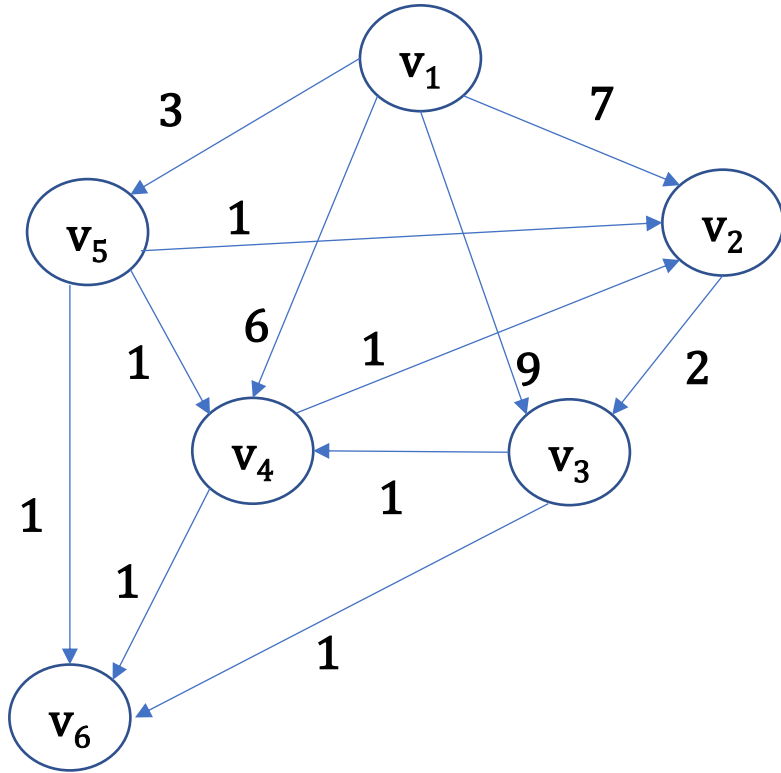
۴- بررسی کار ۴: این کار نمی تواند در مهلت مقرر انجام شود.

۵- بررسی کار ۵: این کار نمی تواند در مهلت مقرر انجام شود.

مقدار نهایی برابر $S=[2,1,3]$ می باشد که دارای بهره $40+35+30=105$ می باشد.

بهره	مهلت	کار
۴۰	۲	۱
۳۵	۱	۲
۳۰	۳	۳
۲۵	۱	۴
۲۰	۲	۵

۱- با استفاده از الگوریتم دایکسترا، کوتاهترین مسیر از V_1 به بقیه رئوس را در گراف زیر پیدا کنید؟



۲- فرضاً ۷ کار با مهلت و بهره مشخص شده داریم. با استفاده از الگوریتم زمانبندی با مهلت، بهره کل را مشخص کنید.

بهره	مهلت	کار
۴۰	۲	۱
۱۵	۴	۲
۶۰	۳	۳
۲۰	۲	۴
۱۰	۳	۵
۴۵	۱	۶
۵۵	۱	۷

۱- برنامه ای بنویسید که مهلت و بهره n کار را از ورودی گرفته و با استفاده از الگوریتم زمانبندی با مهلت معین، زمانبندی بهینه و بهره کل را مشخص کرده و چاپ کنید.

پایان فصل ۳