عنوان درس:

ساختمان داده ها

فصل ۶ : گرافها(Graphs)

مدرس:
اسماعیل نورانی

www.nurani.ir info@nurani.ir

گراف

- اولین بار اولر از گراف برای حل مسائل استفاده کرد.

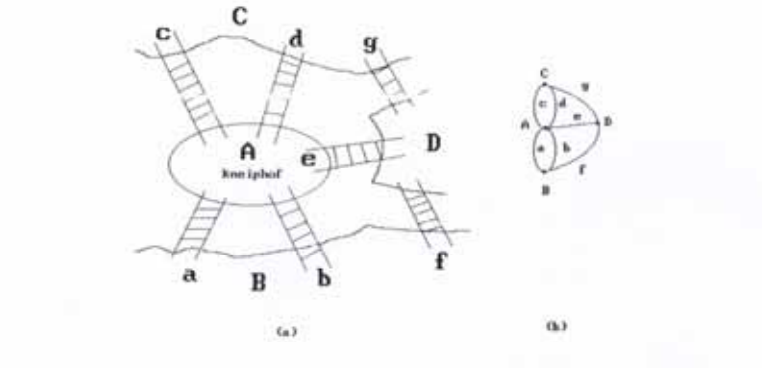


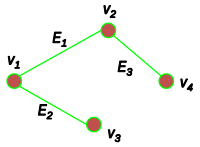
Figure 6.1: The bridges of Koenigsberg

تعریف گراف

- یک گراف G از دو مجموعه تشکیل شده است:
  - مجموعه محدود و غیر تهی از رئوس V(G)
  - مجموعه محدودی از یالها E(G)
- انواع گراف :
  - گراف جهت دار
  - گراف بدون جهت

تعاریف اولیه گرافها

- گراف
  - مجموعه ای از رئوس (Vertex) و لبه ها (Edge) است .
  - در هر دو سر هر لبه گرهی وجود دارد.
- مسیر در گراف
  - دنباله ای از لبه های پشت سر هم بین دو گره
- طول مسیر
  - تعداد یالهای بین دو گره
- حلقه
  - مسیری که راس اول و آخر آن یکی باشد



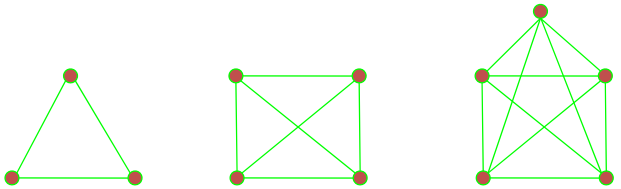
www.nurani.ir info@nurani.ir

تعاریف اولیه گرافها(ادامه)

- گره های همبند
  - اگر از گره **g1** به گره **g2**، مسیری باشد، گره های موجود در این مسیر را گره های همبند گویند.
- گراف همبند
  - اگر در گرافی به ازای هر دو گره انتخابی، مسیری موجود باشد، گراف را گراف همبند گویند

تعاریف اولیه گرافها(ادامه)

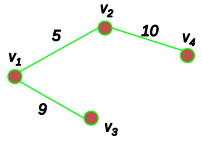
- گراف کامل
  - اگر بین هر دو گره مسیری به طول یک (مسیر مستقیم) باشد، گراف را گراف کامل گویند



- اگر در گراف کامل تعداد **n** راس داشته باشیم، تعداد لبه ها برابر خواهد بود با $n(n-1)/2$

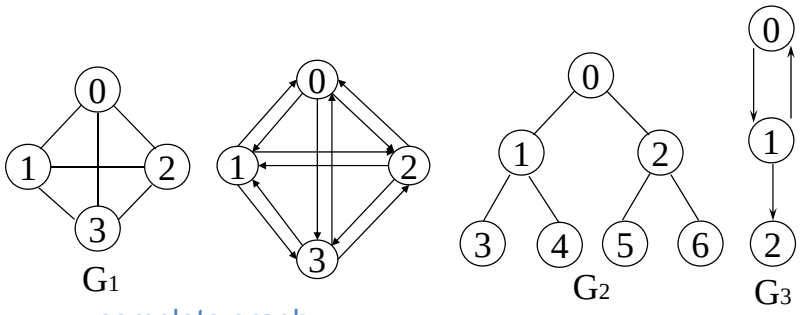
تعاریف اولیه گرافها(ادامه)

- درجه گره
  - تعداد لبه های مرتبط با آن گره
- گراف وزن دار
  - گرافی که در آن به ازای هر لبه، عددی نسبت داده شده باشد



مثالهایی از گراف

- گراف کامل بدون جهت $n(n-1)/2$ یال دارد
- گراف کامل جهت دار $n(n-1)$ یال دارد



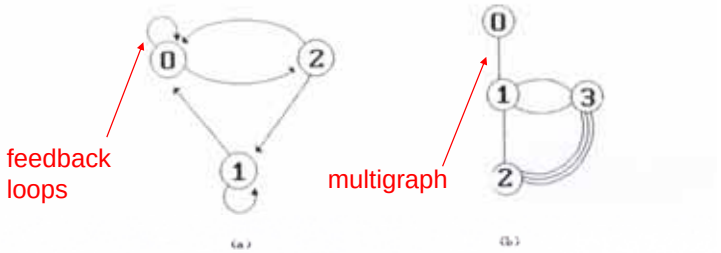
$V(G_1)=\{0,1,2,3\}$ $E(G_1)=\{(0,1),(0,2),(0,3),(1,2),(1,3),(2,3)\}$

$V(G_2)=\{0,1,2,3,4,5,6\}$ $E(G_2)=\{(0,1),(0,2),(1,3),(1,4),(2,5),(2,6)\}$

$V(G_3)=\{0,1,2\}$ $E(G_3)=\{<0,1>,<1,0>,<1,2>\}$

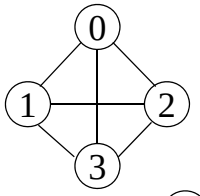
محدودیتهای یک گراف

- در یک گراف نمی توان یالی از یک راس به خود آن راس وصل شود که به آن خود حلقه گفته می شود.
- در یک گراف چند یال بین دو راس یکسان وجود ندارد و اگر این محدودیت برداشته شود به آن گراف چند گانه گفته می شود.

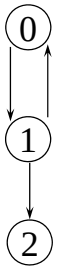


زیر گراف

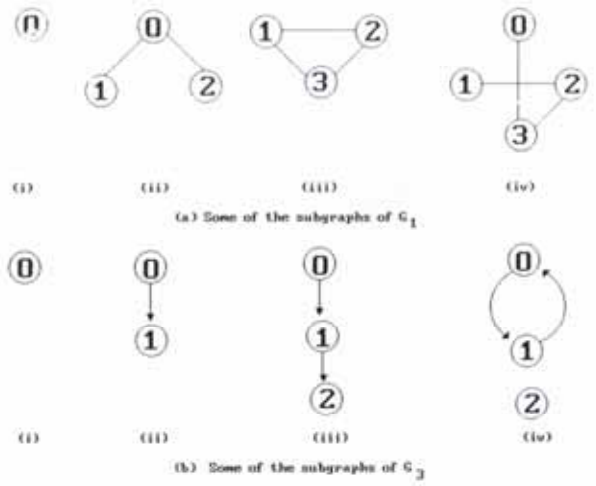
- G' زیر گراف G نامیده می شود اگر :
- $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$.



G1



G3



درجه یک راس

- در یک گراف بدون جهت تعداد یالهای متلاقی با یک راس را درجه آن راس گویند.

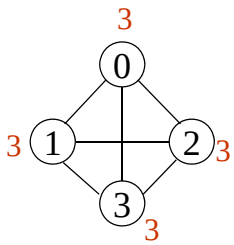
- در یک گراف جهت دار
  - درجه ورودی تعداد یالهای وارد شده به یک راس
  - درجه خروجی تعداد یالهای خارج شونده از یک راس

$$e = \left(\sum_{0}^{n-1} d_i\right) / 2$$

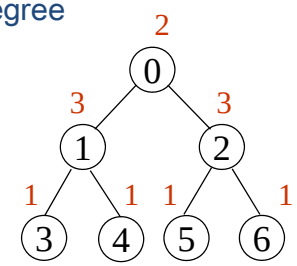
درجه یک راس

undirected graph

degree



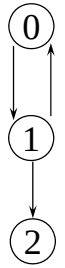
G1



G2

directed graph

in-degree & out-degree



in:1, out: 1

in: 1, out: 2

in: 1, out: 0

G3

محاسبه درجه از روی ماتریس مجاورتی

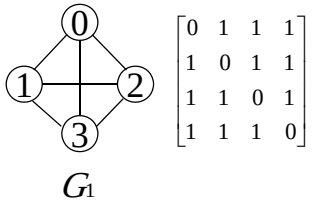
- در گراف بدون جهت درجه یک راس مجموع یک های موجود درسطر متعلق به آن راس

$$\sum_{j=0}^{n-1} adj_mat[i][j]$$

- در گراف جهت دار مجموع سطر درجه خروجی راس و مجموع ستون درجه ورودی آن گراف را نشان میدهد

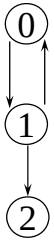
$$ind(vi) = \sum_{j=0}^{n-1} A[j,i] \qquad outd(vi) = \sum_{j=0}^{n-1} A[i,j]$$

نمایش گراف



$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

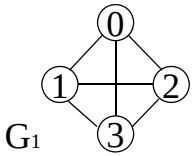
G_1



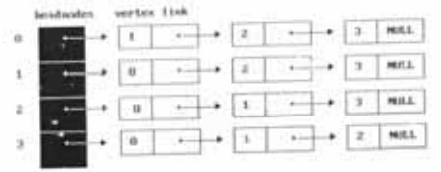
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

G_3

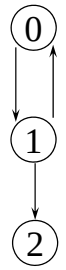
لیست مجاورتی

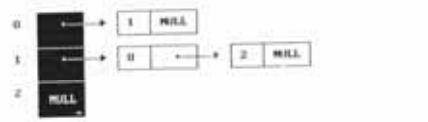


G_1



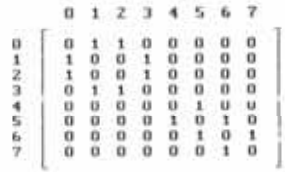
G_3

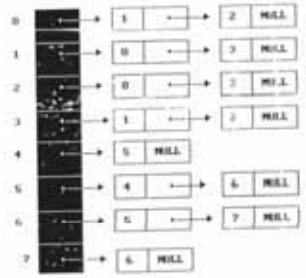




نحوه نمایش گراف

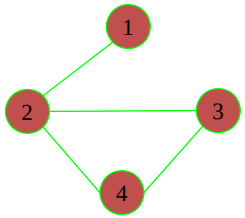
- Adjacency Matrix
- Adjacency Lists





نحوه نمایش گرافها

- ماتریس مجاورت برای گراف بدون جهت

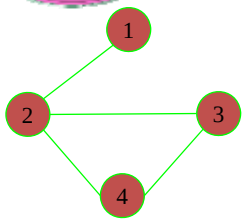


$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

نحوه نمایش گرافها

- ماتریس مجاورت برای گراف بدون جهت

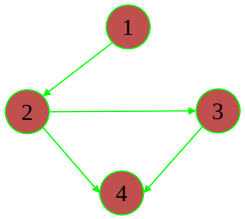
در گراف بدون جهت، ماتریس معادل ماتریسی متقارن می باشد و تعداد یک ها دو برابر تعداد یالها خواهد بود



$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

نحوه نمایش گرافها

- ماتریس مجاورت برای گراف جهت دار

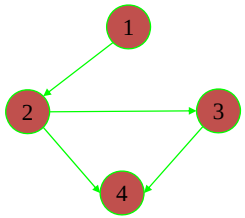


$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

نحوه نمایش گرافها

- ماتریس مجاورت برای گراف جهت دار

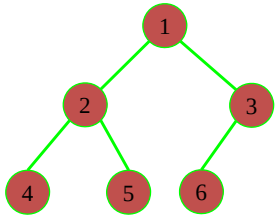
مشکل ماتریس مجاورت این است که با حذف و اضافه کردن رئوس، اندازه ماتریس هم تغییر می کند. و اگر تعدا یالها از گره ها کمتر باشد آنگاه ماتریسمان ماتریس خلوت خواهد شد.

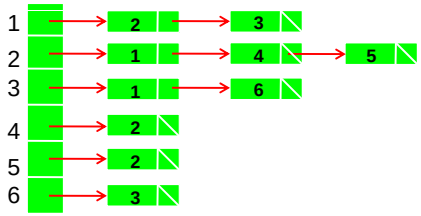


$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

نحوه نمایش گرافها

- لیست مجاورت

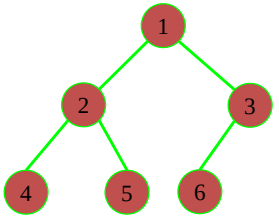


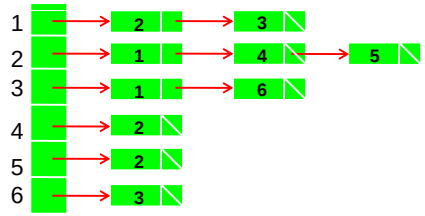


نحوه نمایش گرافها

اگر تعداد یالها از رئوس بیشتر باشد، انگاه روش ماتریس مجاورت بهتر از لیست مجاورت خواهد بود.

- لیست مجاورت

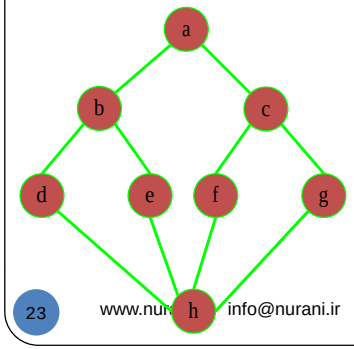




پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

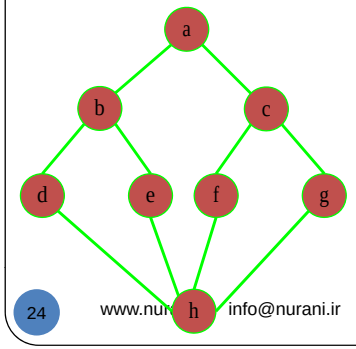
- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود



پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود

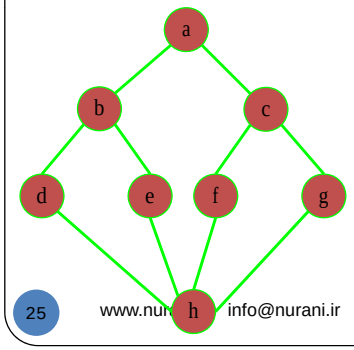


a

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود



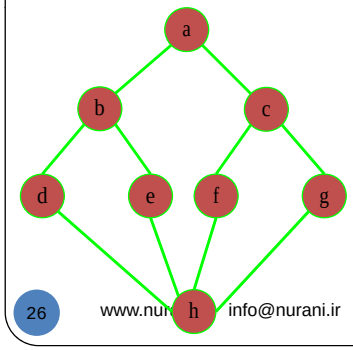
a b

www.nurani.ir info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود



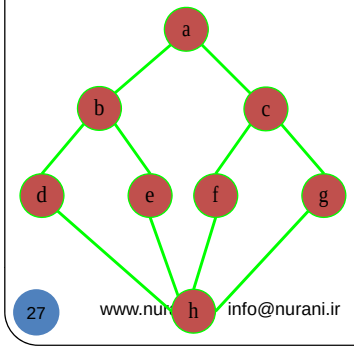
a b d

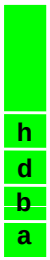
www.nurani.ir info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





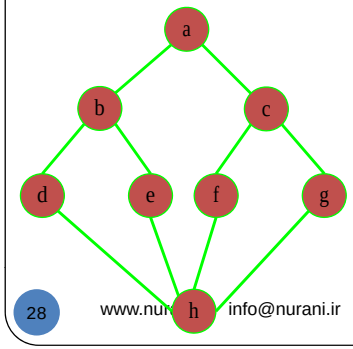
a b d h

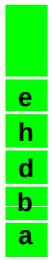
www.nurani.ir info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





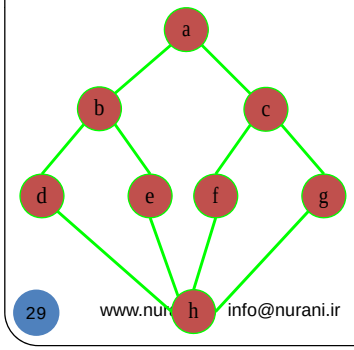
a b d h e

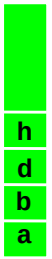
www.nurani.ir info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





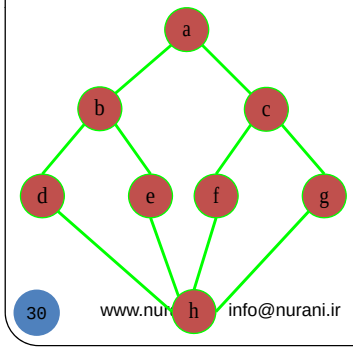
a b d h e

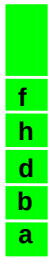
www.nur info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





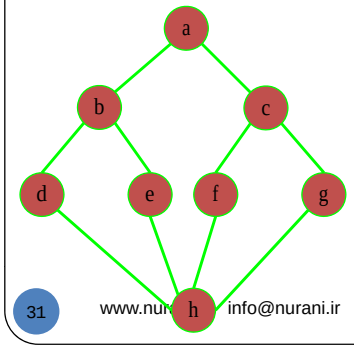
a b d h e f

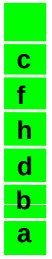
www.nur info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





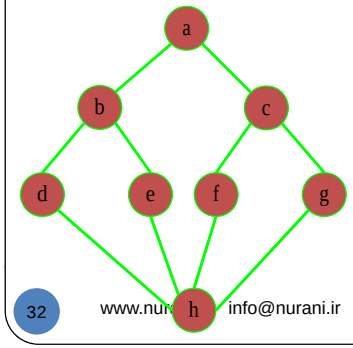
a b d h e f c

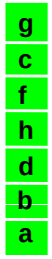
www.nur info@nurani.ir

پیمایش گرافها

با استفاده از پشته پیاده سازی می شود.

- پیمایش عمقی **dfs**
  - از گره اول شروع می کنیم.
  - فرزند اول را وارد پشت کرده و پیمایش می شود.
  - فرزند اول عنصر جدید وارد شده و پیمایش می شود.
  - اگر تمام فرزندان گرها ملاقات شود آن گره **pop** می شود





a b d h e f c g

www.nur info@nurani.ir

پیمایش گرافها

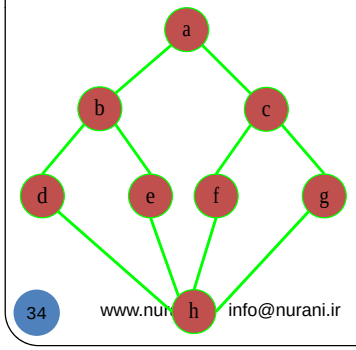
- پیمایش عمقی **dfs**

```
dfs( v: vertex)
  {
      process(v.data)
      visited[v]=true
      for (هر فرزند ملاقات نشده v)
        if(اگر گره قبلا پیمایش نشده)
            dfs(گره)
  }
```


پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

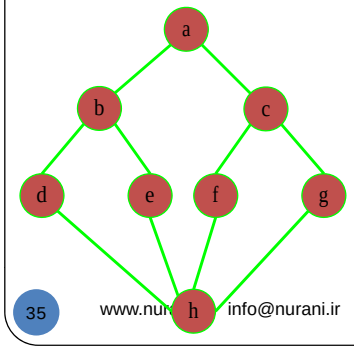
- پیمایش سطحی **bfs** (**Breatch First Search**)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی **bfs** (**Breatch First Search**)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.

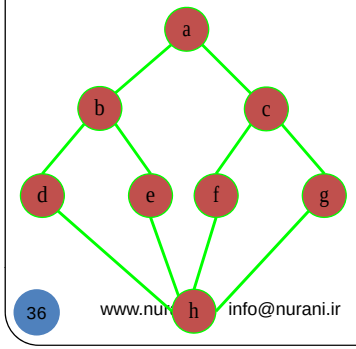


a

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی **bfs** (**Breatch First Search**)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.

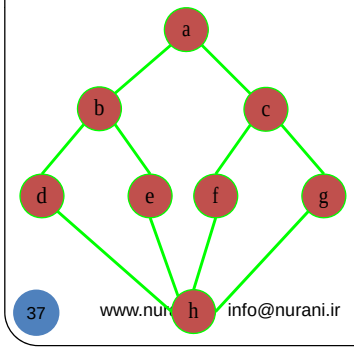


a

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



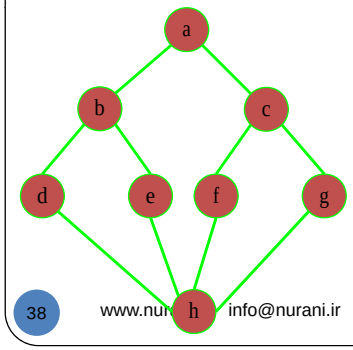
b	c					

a b c

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



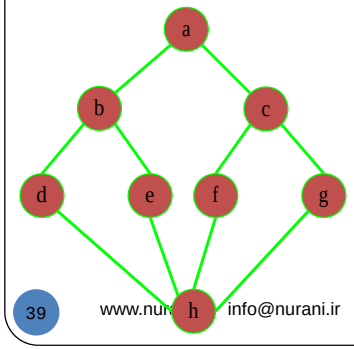
c						

a b c

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



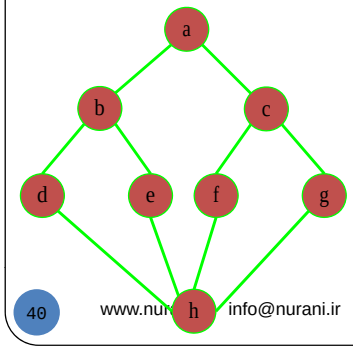
c	d	e				

a b c d e

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



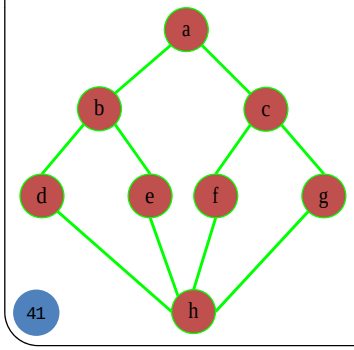
d	e					

a b c d e

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



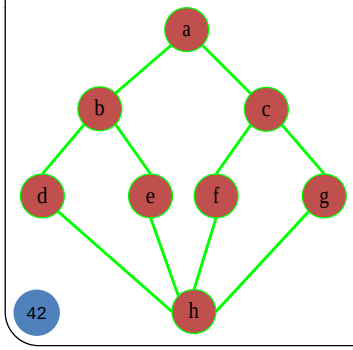
d e f g

a b c d e f g

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



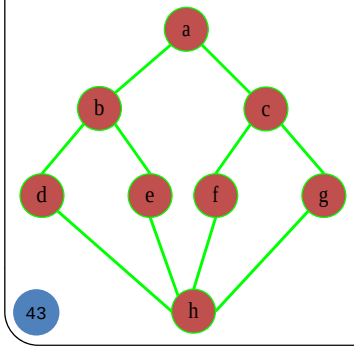
e f g

a b c d e f g

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



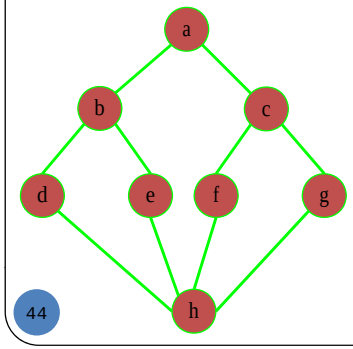
e f g h

a b c d e f g h

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



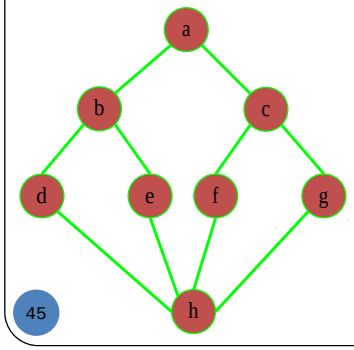
f g h

a b c d e f g h

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



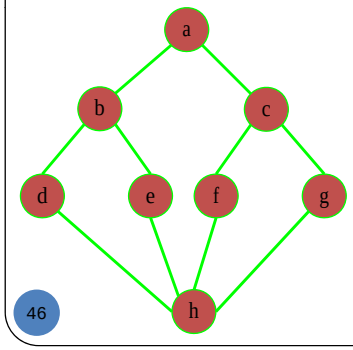
g h

a b c d e f g h

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



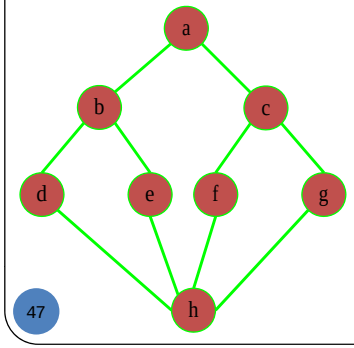
h

a b c d e f g h

پیمایش گرافها

با استفاده از صف پیاده سازی می شود.

- پیمایش سطحی bfs (Breatch First Search)
  - گره اول را وارد صف می کنیم
  - حال گره اول را از صف حذف و همه فرزندان آن را وارد صف می کنیم.
  - حا لبه ترتیب یکی یکی از صف حذف کرده و فرزندان عنصر حذف شده را وارد صف می کنیم.



a b c d e f g h

پیمایش گرافها

- پیمایش سطحی bfs (Breatch First Search)

```
bfs( v: vertex)
  {
        visited[v]=true
        addq(q,v)
        while (not emptyqueue (q) )
          {
              delq();
              for (all node w adjacent v )
                {
                    addq(q,w)
                    visited[w]=true   }   }   }
```

درختهای پوشا با حداقل هزینه

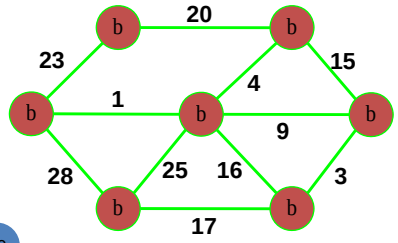
- درخت پوشا درختی است که تمام گره های موجود را با تعدادی از یالها شامل شود.
- درخت پوشایی که کمترین مجموع وزن یال را داشته باشد، درخت پوشای مینیمم گویند.
- الگوریتم های ایجاد درخت پوشای مینیمم
  - الگوریتم راشال
  - الوریتم پریم

www.nurani.ir info@nurani.ir

درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

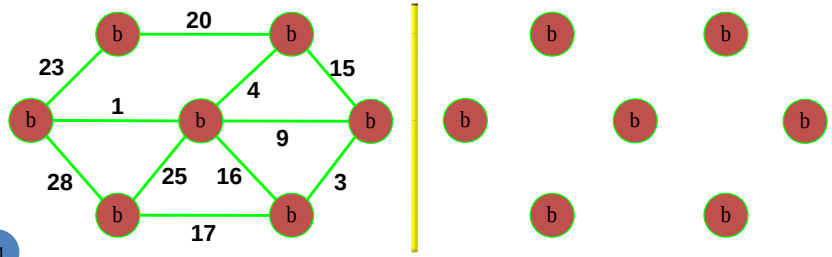
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

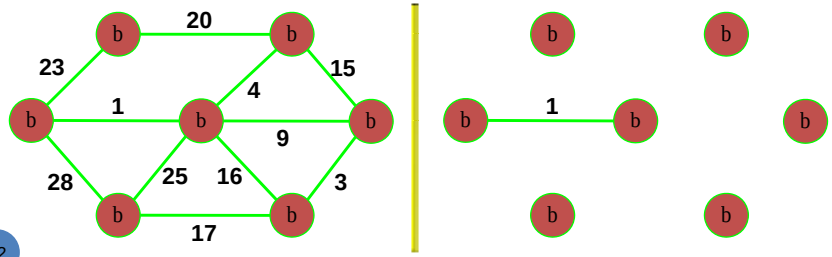
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

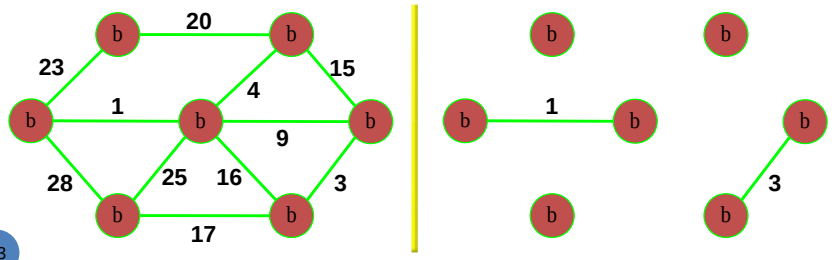
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

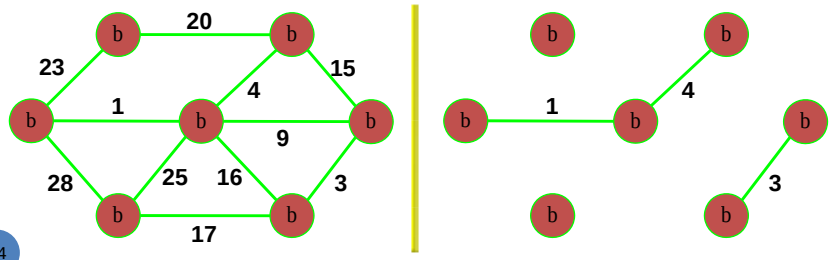
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

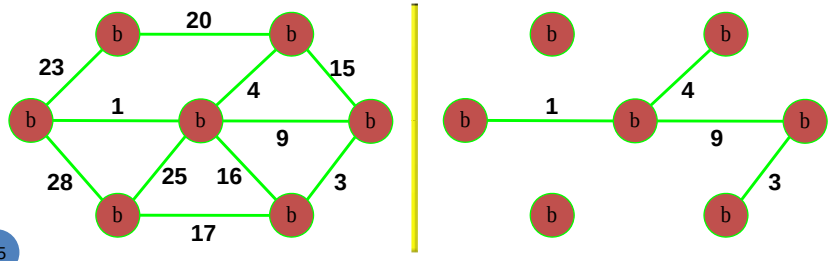
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

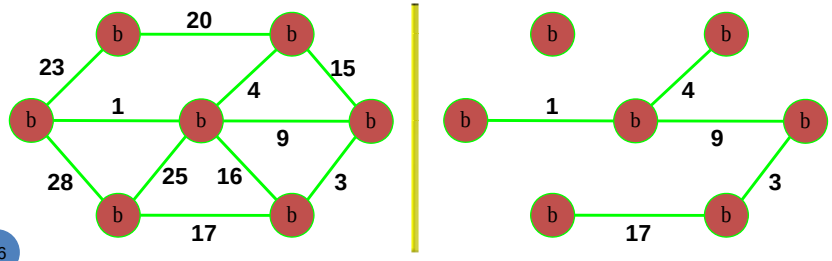
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

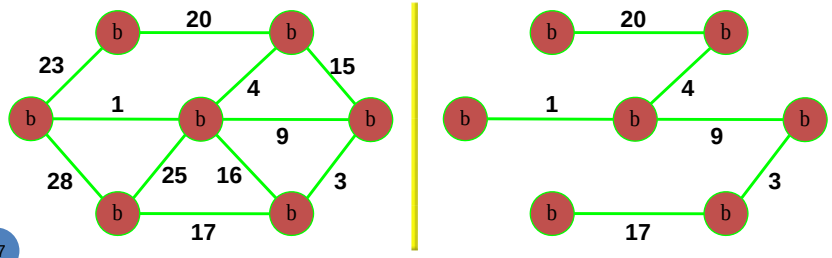
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

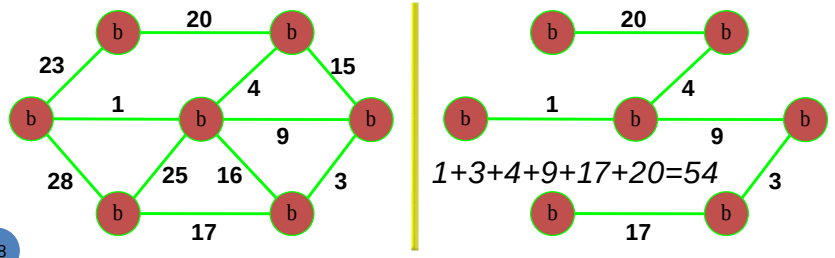
در این روش در هر مرحله با یک جنگل سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم راشال
  - ابتدا تمام گره ها را بدون یالها رسم می کنیم
  - حال یکی یکی یالها را از وزن کوچک به بزرگ طوری اضافه می کنیم که ایجاد حلقه نکند.

در این روش در هر مرحله با یک جنگل سرو کار داریم

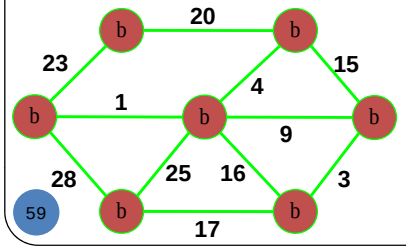


1+3+4+9+17+20=54

درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های اضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

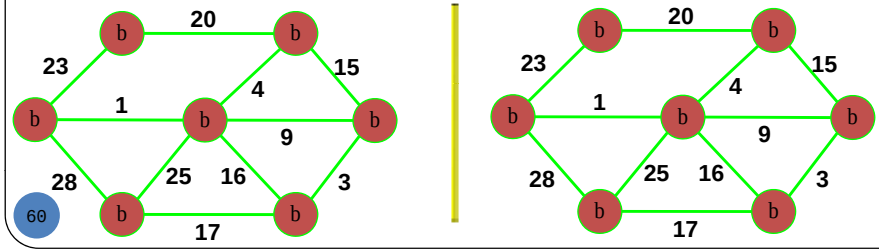
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های اضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

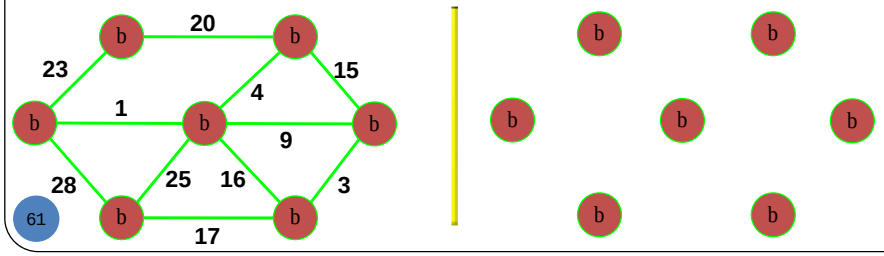
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

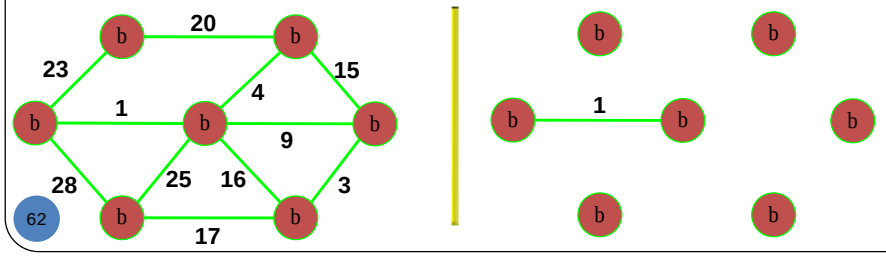
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

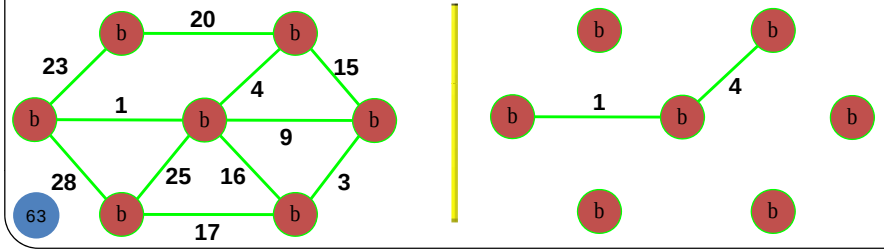
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

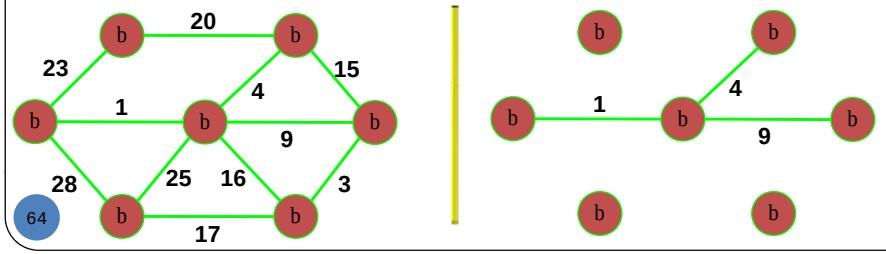
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

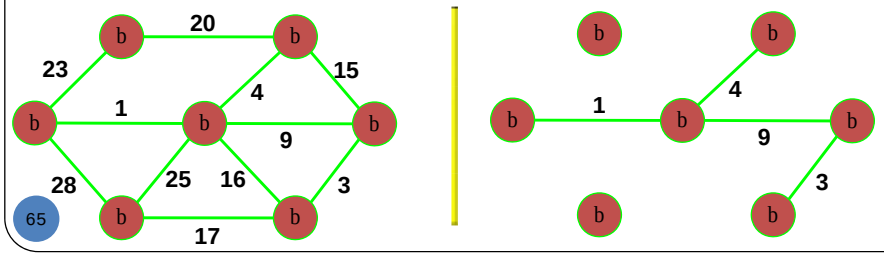
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

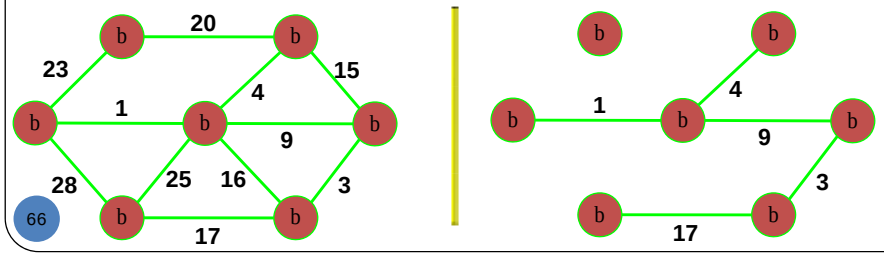
در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

در این روش در هر مرحله با یک درخت سرو کار داریم



درختهای پوشا با حداقل هزینه

- الگوریتم پریم
  - از یک یال شروع می کنیم
  - کوچکترین یال متصل به آن را وصل می کنیم
  - حال در بین یال های مجاور یال های ضافه شده، کوچکترین یال را پیدا کرده و اضافه می کنیم

در این روش در هر مرحله با یک درخت سرو کار داریم