

گراف ها

7-1 مقدمه

علیرغم این که اولین مقاله در زمینه نظریه گراف در سال 1736 منتشر شد و چند نتیجه قابل توجه و مهم در نظریه گراف در قرن هجدهم به دست آمد، اما تنها از دهه ۱۹۲۰ بود که این نظریه به صورت گسترده و عمیق مورد توجه قرار گرفت. بی شک، یکی از دلایل توجه به نظریه گراف، قابلیت استفاده از آن در عرصه های متعدد از جمله علوم کامپیوتر، شیمی، تحقیق در عملیات مهندسی برق، زبان شناسی، اقتصاد و ... است. درخت ها که در فصل پیش با آن ها آشنا شدیم، نوعی خاص از گراف هستند. در این فصل به نظریه مقدماتی گراف و معرفی برخی از اکاربردهای آن می پردازیم.

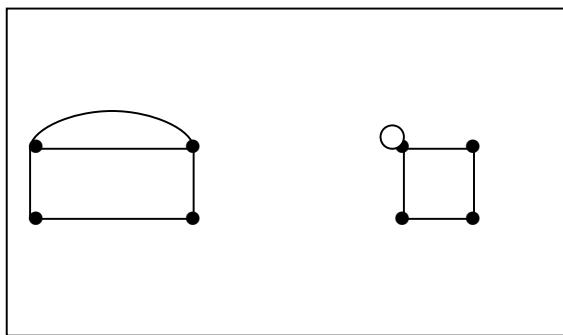
7-2 گراف ساده / چندگانه

فرض کنید V یک مجموعه متناهی و ناتهی باشد و E مجموعه ای باشد که عناصر آن زیرمجموعه های دو عنصری از عناصر متمایز V باشند، در این صورت به زوج مرتب (V,E) یک گراف ساده می کنیم. عناصر V را رئوس و تعداد عناصر V یعنی $|V|$ را مرتبه می نامیم، همچنین، عناصر E را یال (به) و تعداد عناصر آن یعنی $|E|$ را اندازه گراف می نامیم.

نکته : در یک گراف مجموعه رئوس نمی تواند تهی باشد ، ولی مجموعه یال ها می تواند تهی باشد، که در این صورت گراف مذبور را گراف تهی می نامیم .

نکته : اگر $\{a,b\}$ یالی از گراف G باشد ، می گوییم a,b دو رأس مجاور یا دو سر یک یال از گراف هستند .

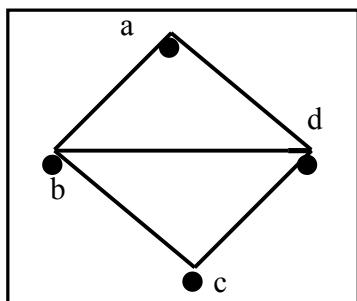
گراف چندگانه ، گرافی است که شامل طوقه با یال های چندگانه است . مقصود از طوقه یالی است که دو انتهایش بر هم منطبق است که دو انتهایش برهم منطبق است و اگر دو رأس با بیش از یک یال به هم متصل باشند ، این یال ها را چندگانه می گوییم . شکل مقابل ، نمونه ای از گرافهای چندگانه را نشان می دهد .



نکته : تعداد گراف های ساده n رأسی برابر است $t = \frac{n(n-1)}{2}$. توجه کنید که تعداد یال های یک گراف n رأسی حداقل و حداقل $\frac{n(n-1)}{2}$ است .

7-3. درجه یک رأس

تعداد یال هایی که از یک رأس می گذرد ، درجه آن رأس نامیده می شود. درجه رأس a را با $\deg(a)$ نشان می دهیم و چنانچه $\deg(a)$ عددی فرد باشد ، رأس را فرد و در غیر اینصورت رأس را زوج می نامیم .

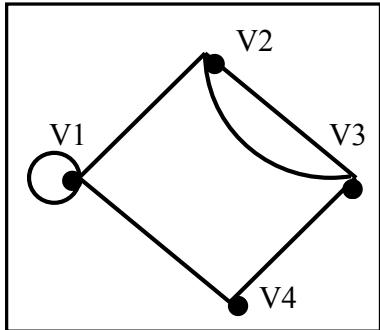


$$\deg(a) = \deg(c) = 2$$

$$\deg(b) = \deg(d) = 3$$

نکته : در گراف های چندگانه برای محاسبه درجه یک رأس طوقه را دو یال محسوب کنید .

✓ تست



درجه رأس V1 در گراف زیر چند است ؟

الف. 4. ۵ ب. 3. ۲ ج. 1. ۲

حل : گزینه د.

✓ تست

اگر $V = \{a, b, c, d\}$ چند گراف ساده وجود دارد که $\deg(c) = \deg(d) = 3$ و $\deg(b) = \deg(a) = 1$

الف. ۰. ۱ ب. ۲^۶ ج. ۱. ۰

حل : گزینه د. چنین گرافی وجود ندارد . زیرا $\deg(c) = 3$ و سه یال از c به a و b و d باید وجود داشته باشد و

$\deg(c) = 3$ و سه یال از d به a و b و c باید داشته باشیم و در نتیجه :

$$\text{Deg}(a) \geq 2, \text{ Deg}(b) \geq 2$$

که خلاف فرض است . لذا چنین گرافی وجود ندارد .

نکته ها :

1. در هر گراف ، مجموع درجات رئوس دو برابر یال ها است . یعنی :

$$\sum_{i=1}^p \deg Vi = 2q$$

که p مرتبه و q اندازه گراف است (این مطلب به لم دست دادن مشهور است) .

2. اگر δ برابر مینیمم درجات رئوس و Δ برابر ماکزیمم درجات رئوس باشد ، آنگاه :

3. تعداد رئوس فرد در هر گراف عددی زوج است .

۷ تست

کدام یک از دنباله های زیر می تواند دنباله درجه های رئوس یک گراف ساده باشد؟

ب. ۰،۲،۳،۳،۴،۴،۴،۳،۲،۲،۱،۰

الف. ۱،۰،۲،۲،۳،۴،۴،۴،۳،۲،۱،۰

د. ۰،۰،۰،۲،۲،۳،۳،۳،۳

ج. ۰،۱،۰،۲،۳،۳،۶،۶

حل : گزینه د . مجموع جملات داده شده در گزینه های الف و ج عددی فرد است و لذا این گزینه ها نادرست هستند. در گزینه ج چون یک رأس درجه ۶ داریم ، رأس درجه صفر (ایزوله) نمی توانیم داشته باشیم .

نکته : یک گراف ساده از مرتبه n نمی تواند دو رأس از درجه $n-1$ و یک رأس از درجه ۱ داشته باشد.

7-4. گراف کامل

گرافی که حداکثر یال های ممکن را دارد و هر دو رأس با یک یال به هم متصل هستند ، گراف کامل نامیده می شود.

گراف کامل n رأسی را با k_n نشان می دهیم .

نکته ها

۱. تعداد یال های k_n برابر $\binom{n}{2} = \frac{n(n-1)}{2}$ است .

۲. درجه هر رأس در k_n برابر $n-1$ است.

✓ تست

گراف کامل مرتبه 7 چند یال دارد ؟

الف. 21 ب. 15 ج. 7 د. 6

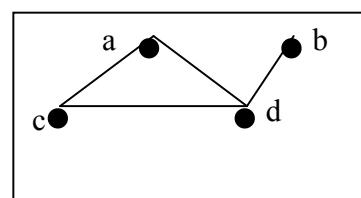
حل : گزینه الف . با توجه به نکته فوق $q = (7 \times 6) / 2 = 21$ یال دارد .

7-5. مکمل یک گراف

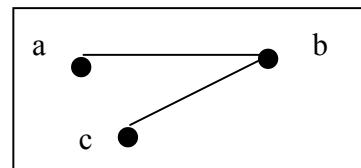
اگر G یک گراف ساده باشد ، مکمل G با \bar{G} نشان داده می شود ، گرافی است که رئوس آن همان رئوس G و یال های ان از وصل کردن رأس های به هم متصل نشده در گراف G به دست می آیند .

مثال

مکمل گراف زیر را رسم کنید .



حل : تعداد یال های G و \bar{G} روی هم برابر $n(n-1)/2$ است .



نکته : اگر $q(G)$ و $q(\bar{G})$ به ترتیب اندازه های G و \bar{G} باشند ، و G دارای n رأس باشد ، آن گاه :

$$q(G) + q(\bar{G}) = \binom{n}{2}$$

✓ تست

فرض کنید G یک گراف ساده با n رأس باشد. اگر G ، 56 یال و \bar{G} دارای 80 یال باشد ، مقدار n کدام است ؟

- الف. 16 ب. 17 ج. 18 د. 19

حل : گزینه ب.

$$80+56=136 \Rightarrow n(n-1)=272 \Rightarrow n=17$$

✓ تست

اگر گراف \bar{G} مکمل گراف (V,E) باشد ، کدام یک از عبارات زیر صحیح است ؟

الف. $\bar{G} = (\bar{V}, E)$ ب. $\bar{G} = (V, \bar{E})$

ج. $\bar{G} = (V, E)$ د. $\bar{G} = (\bar{V}, \bar{E})$

حل : گزینه ج. رئوس G و \bar{G} یکسان است و مجموعه یال های آن ها مکمل یکدیگر است .

7-6. گراف های k منتظم

اگر درجه هر رأس یک گراف برابر k منتظم می نامیم . هر گراف کامل k_n یک گراف $n-1$ منتظم مرتبه n است .

نکته : در گراف r منتظم مرتبه n داریم :

$$q = \frac{nr}{2} \quad (\text{تعداد یال ها})$$

لذا n و r هر دو نمی توانند فرد باشند.

✓ تست

فرض کنید $G = (V, E)$ دارای 10 یال و 2 رأس از درجه 3 باشند، تعداد رئوس این گراف برابر است با :

- الف . 4 ب . 5 ج . 6 7.د

حل : گزینه ج. اگر تعداد رئوس درجه 3 برابر n باشد ، مجموع درجات رئوس برابر است با :

$$2 \times 4 + 3 \times n = 8 + 3n = 20$$

$$\therefore n = 4$$

7-7. مسیر / دور

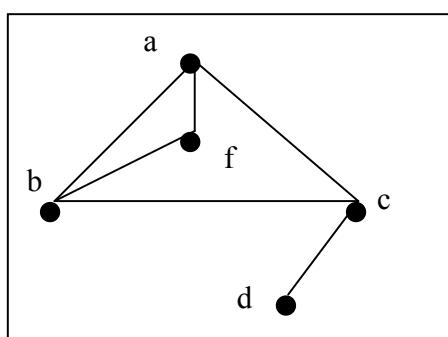
در گراف $G = (V, E)$ یک مسیر از V_0 به V_n به دنباله ای از رئوس متمایز به صورت $V_0, V_1, V_2, \dots, V_n$ است

که هر دو رأس متواالی مجاورند. طول این مسیر برابر n تعریف می شود (طول مسیر همان تعداد یال های مسیر است).

نکته : هر دنباله به طول یک از رئوس ، مسیری با طول صفر است.

مسیری که ابتدا و انتهای یکسان داشته باشد ، دور نامیده می شود .

مثال



a - b - c - a دور از a به a

a - b - f - c - a مسیرهای با طول 2 از a به a

b - c - d مسیری از b به d

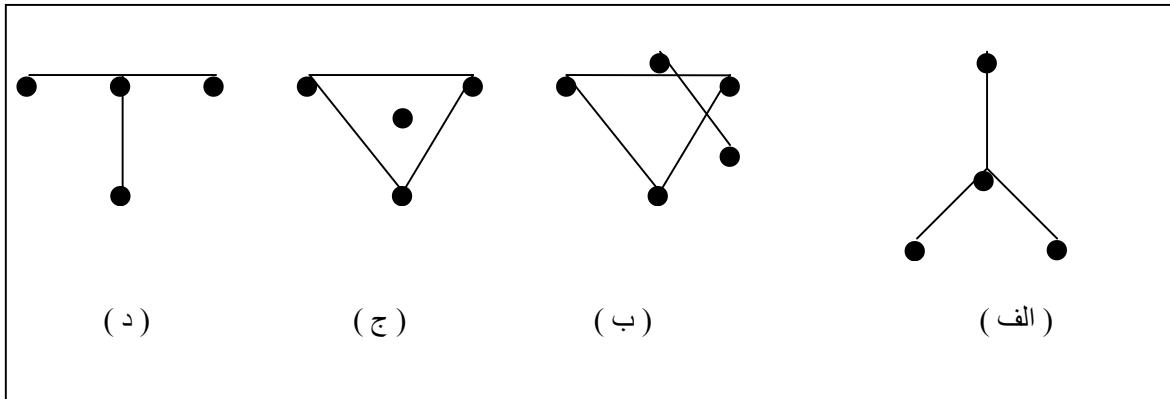
7-8. گراف همبند (پیوسته)

گراف G را همبند می گوییم ، هر گاه بین هر دو رأس متمایز آن حداقل یک مسیر وجود داشته باشد ، به گرافی که همبند نیست ناهمبند می گوییم .

مثال

چند تا از گراف های زیر همبند می باشند ؟

- الف. 1. 2. ب. 3. ج. 4. د.



حل : گزینه ب. گراف ج دارای رأس درجه صفر (ایزوله یا تنها) است و همبند نیست . گراف ب از دو قسمت مجزا تشکیل شده و همبند نیست .

نکته : اگر یک گراف ساده حداقل دو رأس داشته باشیم ، آن گاه حداقل دو رأس هم درجه داریم .

✓ تست

اگر $G = (V, E)$ یک گراف ساده و همبند باشد که $q = 17$ یال دارد و درجه رأس آن حداقل 3 است، آن گاه

G حداقل چند رأس دارد؟

- الف. 10 ب. 11 ج. 12 د. نمی توان تعیین کرد

حل : گزینه ب . مجموع درجات رئوس حداقل برابر $3n$ است ، لذا $3n \leq 2q \geq 3n$ و در نتیجه

7-9. گراف اویلری

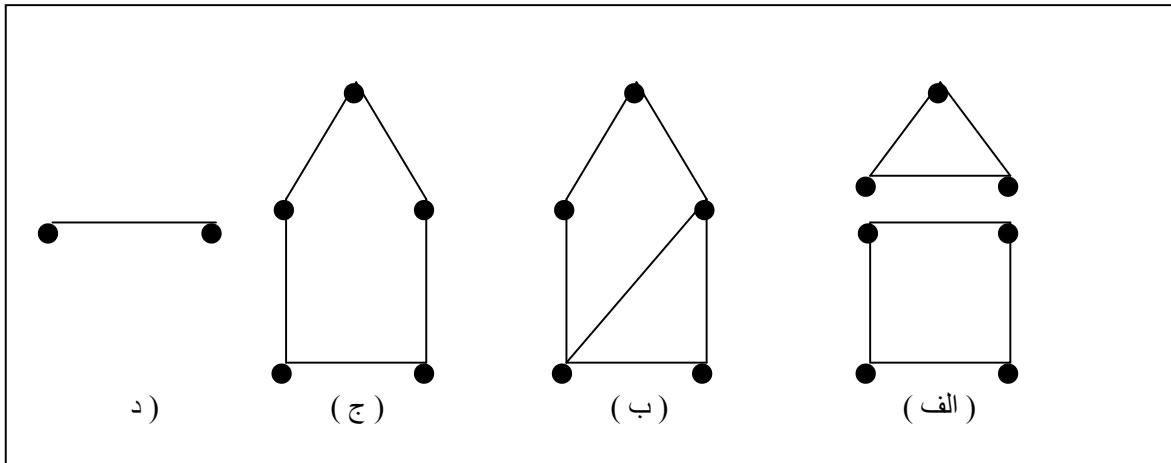
مسیری که از همه یال های یک گراف دقیقا یک بار بگذرد ، مسیر اویلری نامیده می شود و هر گراف شامل دور اویلری ، گراف اویلری نامیده می شود.

نکته : دو شرط لازم و کافی برای این که یک گراف اویلری باشد ، عبارتند از :

1. همبند بودن گراف
2. زوج بودن همه درجات رئوس گراف

✓ تست

کدام یک از گراف های زیر اویلری است ؟



حل : گزینه ج. توجه کنید که گراف گزینه د فاقد دور است ، گراف گزینه الف همبند نیست و گراف گزینه ب دارای رئوس درجه فرد است .

7-10. گراف همیلتونی

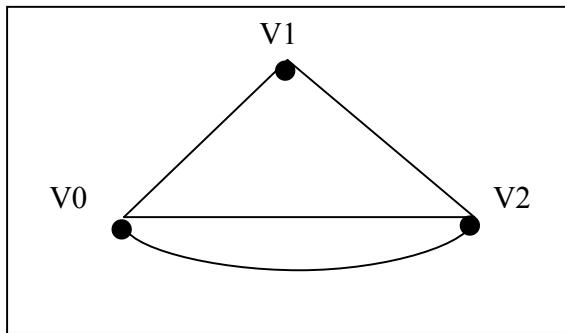
در یک گراف به دوری که شامل همه رأس ها باشد ، دور همیلتونی می گویند ، یک گراف را همیلتونی می نامیم هرگاه شامل یک دور همیلتونی باشد. توجه کنید که هر گراف همیلتونی همبند بوده و حداقل 3 رأس دارد .

نکته: اگر G یک گراف ساده از مرتبه p باشد و $p \geq 3$ و $\delta \geq \frac{p}{2}$ آن گاه G همیلتونی است .

نکته: هر گراف کامل همیلتونی است.

✓ تست

در مورد گراف شکل مقابل کدام گزینه صحیح می باشد ؟



الف . فاقد دور همیلتونی است .

ب . دارای دور همیلتونی است ولی دور اویلری ندارد.

ج . فاقد دور اویلری و دور همیلتونی است .

د. دارای دور اویلری است.

حل : گزینه ب صحیح است . چون $V_0V_1V_2V_0$ دور همیلتونی است و دور اویلری در این گراف نداریم (درجه V_0

و V_2 فرد است).

7-11. ماتریس همجواری (مجاورت)

فرض کنید $G=(V,E)$ یک گراف ساده از مرتبه n باشد ، ماتریس مجاورت گراف G که با $A(G)$ نشان داده می

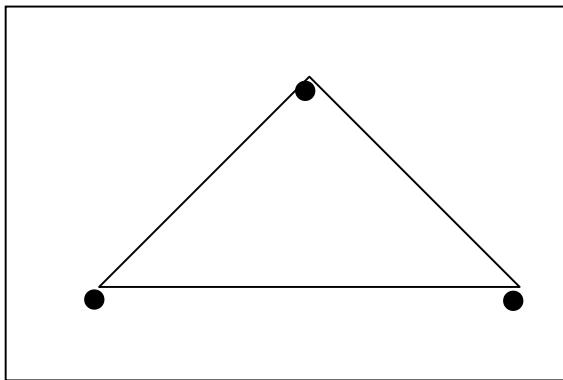
شود ، به صورت زیر تعریف می گردد:

$$A(G) = [a_{ij}]_{n \times n}$$

که در آن V_i و V_j مجاور باشند ، $a_{ij} = 1$ در غیر اینصورت 0

مثال

ماتریس مجاورت گراف مقابل عبارت است از :



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

ویژگی های ماتریس مجاورت یک گراف ساده به شرح زیر است :

1. متقارن است .
2. عناصر قطر اصلی آن صفر هستند.
3. مجموع عناصر آن دو برابر تعداد یال ها است .
4. مجموع عناصر هر سطر (یا ستون) با درجه رأس متناظر آن برابر است .

نکته : اگر A ماتریس مجاورت G باشد ، عناصر اصلی A^2 برابر درجات رئوس G هستند و در A^k درایه (j,i) ام برابر

تعداد مسیرهای با طول K بین دو رأس j و i است .

تست ✓

فرض کنید A ماتریس مجاورت گراف کامل K_5 است . مجموع درایه های قطر اصلی A^2 کدام است ؟

- | | | | |
|-------|-------|-------|---------|
| د. 30 | ج. 15 | ب. 20 | الف. 25 |
|-------|-------|-------|---------|

حل : گزینه ب. مجموع درجات رئوس k_n برابر A^2 است .

{ simple graph drawing}

Program draw;

Uses graph;

var

Gd,gm,I,j,n:integer;

X:array [1..10]of integer;

y:array [1..10]of integer;

m:array [1..10,1..10]of integer;

begin

readln(n);

for i:=1 to n do

begin

for j:=1 to i-1 do

begin

read(m [i,j]);

m[j,i]:=m[i,j];

end;

m[i,i]:=0;

end;

gd:=DETECT;

initgraph(gd,gm,'c:\tp\bgi');

if graphresult<>grok then

begin

writeln(' hallo! Graphics error heda ');

readln;

exit;

end;

ساختمان داده ها- آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

```
setbkcolor(6);
setcolor(7);
randomize;
for i:=1 to n do
begin
  x[i]:=random(getmaxx);
  y[i]:=random(getmaxy);
  putpixel(x[i],y[i], 4);
end;
for i:=1 to n do
  for j:=1 to i-1 do
    if m[i,j]=1 then
      line(x[i],y[i],x[j],y[j]);
readln;
closegraph;
end.
```

7-12. گراف جهت دار (دیاگراف)

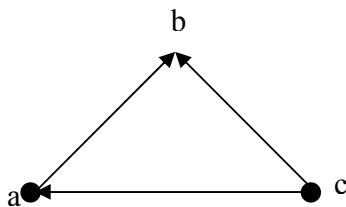
ه گراف $G = (V, E)$ که در آن V مجموعه ای متناهی و ناتهی از رئوس است و E زیرمجموعه ای از $V \times V$ است ،

یک گراف جهت دار می گوییم . در یک گراف ممکن است برخی از یال ها دارای جهت باشند و برخی نباشند که در

این صورت گراف را مختلط می نامیم .

در یک گراف جهت دار یال های متصل به هر راس دو دسته اند:

1. یال های خروجی از رأس 2. یال های ورودی به رأس

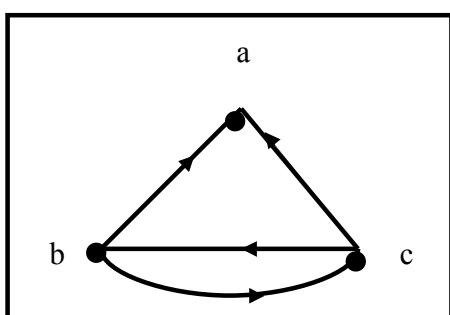


لذا دو نوع درجه برای یک رأس مانند a تعریف می کنیم . outdeg یا درجه خروجی و indeg یا درجه ورودی رأس .

نکته : حداکثر تعداد یال ها (لبه ها) در یک گراف جهت دار n رأسی برابر $n(n-1)$ است .

مثال

در گراف مقابل داریم :



$$\text{Indeg}(a) = 2 \quad \text{outdeg}(a) = 0$$

$$\text{Indeg}(c) = 1 \quad \text{outdeg}(c) = 2$$

$$\text{Indeg}(b) = 1 \quad \text{outdeg}(b) = 2$$

توجه : هر درخت یک گراف همبند بدون دور است . درخت هایی که در فصل قبل مورد بررسی قرار دادیم ، درخت های جهت دار بودند . در هر درخت مرتبه $n-1$ اندازه برابر n است و بین هر دو رأس متمایز در درخت دقیقا یک مسیر وجود دارد (تعداد یالها در درخت یک واحد از تعداد راسها کمتر است)

✓ تست

گراف همبند G فاقد دور است . مجموع مرتبه و اندازه آن کدام عدد می تواند باشد ؟

د. 20

ج. 18

ب. 15

الف. 12

حل : گزینه ب . چنین گرافی یک درخت است و لذا چنانچه اندازه $n-1$ باشد ، مرتبه n است و در نتیجه

$$p + q = n + (n-1) = 2n - 1$$

یعنی مجموع اندازه و مرتبه عددی فرد است .

7-13. پیمایش گراف

اگر $(V,E) = G$ یک گراف و X گره ای در این گراف باشد ، در پیمایش گراف G لازم است مشخص کنیم چه گره

هایی از طریق گره X قابل دستیابی اند . برای این کار از دو روش استاندارد استفاده می شود که به جست و جوی عمقی (

پیمایش عمقی) و جست و جوی عرضی (پیمایش عرضی) معروفند . الگوریتم این پیمایش ها بر اساس نشانگر گره ها

نوشته می شود . نشانگر می تواند مقادیر مختلفی را بپذیرد تا ت Shan دهنده وضعیت گره باشد . گره می تواند یکی از

شرایط زیر را داشته باشد :

1. حالت آماده (گره آماده است تا دستیابی شود) .

2. حالت انتظار ، از گره عبور شد و منتظر است تا پردازش شود .

3. ملاقات شده

هر یک از این حالات را می توان با یک مقدار مشخص کرد . به عنوان مثال ، حالت آماده با 1 ، حالت انتظار با 2 و حالت ملاقات شده با 3 مشخص می شود .

الگوریتم جست و جوی عمقی (dfs)

1. فیلد نشانگر تمام گره ها را برابر 1 قرار دهید که به معنای آماده پیمایش است .

2. گره شروع (مثلا A) را پشته قرار دهید و فیلد نشانگر آن را برابر با 2 قرار دهید که به معنای انتظار ملاقات است .

3. مراحل 4 و 5 را آن قدر ادامه دهید تا پشته خالی شود .

4. گره موجود در بالای پشته را pop کنید و در خروجی بنویسید و فیلد نشانگر آن را برابر با 3 قرار دهید که به معنای آن است که ملاقات شده است . این گره را p می نامیم .

5. تمام گره های همچوار p را که وضعیت آن ها در حالت آماده است (یعنی قبل از پشته قرار نگرفته اند) در پشته قرار دهید و وضعیت آن ها را به حالت 2 تغییر دهید که به معنای انتظار است .

به بیان دیگر ، الگوریتم dfs را می توان به صورت زیر نوشت :

Procedure dfs (V : integer);

Begin

 Writeln(V^.data) :

 Visited[V] := true :

 For (each vertex w adjacent to V) do

 If (not visited [W]) then

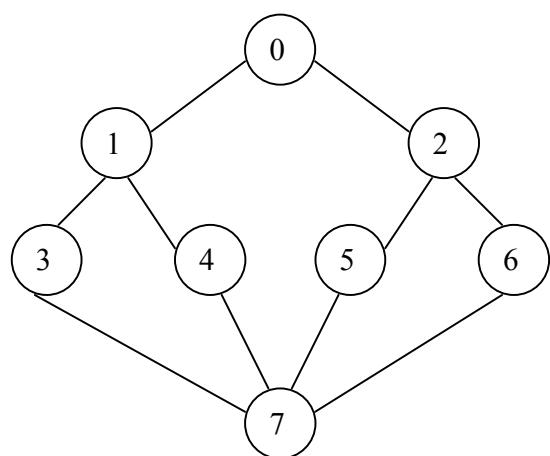
 Dfs(W) :

End :

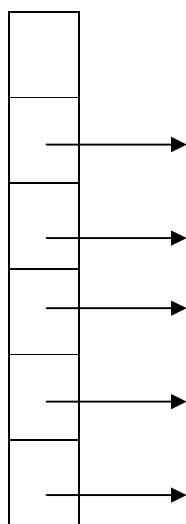
نکته: اگر گراف G توسط لیست های مجاورتی ارائه شود زمان جست و جوی dfs برابر $O(q)$ است و چنانچه با ماتریس مجاورت نمایش داده شود، زمان اجرا برابر $O(n^2)$ است.

مثال

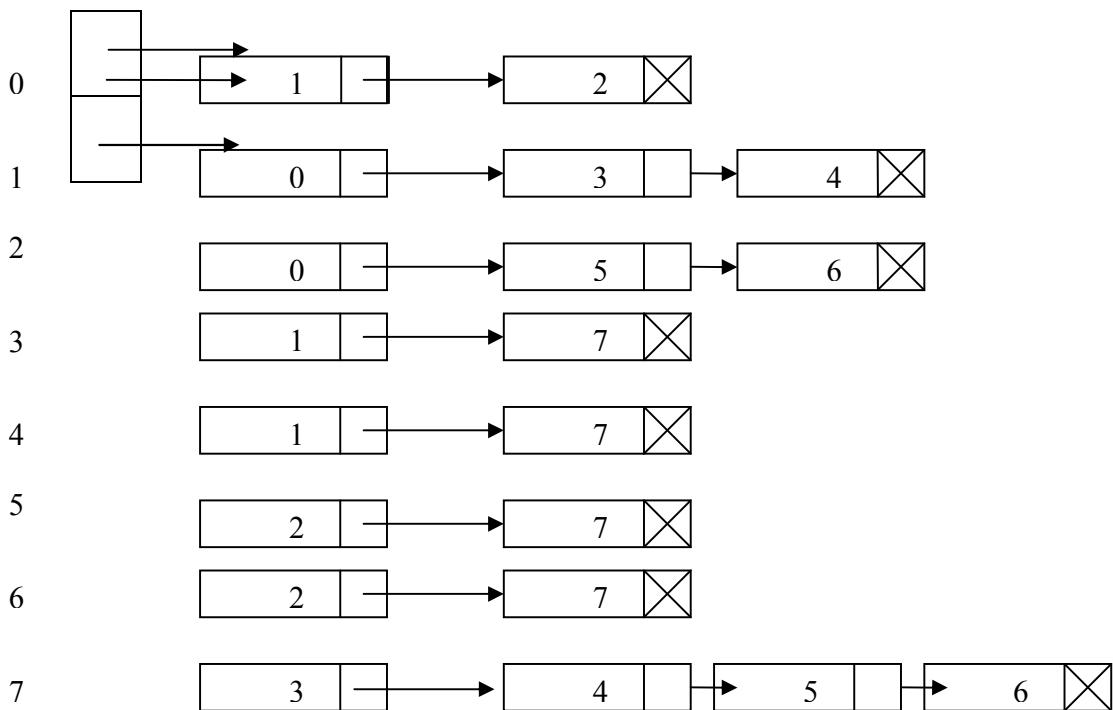
پیمایش عمقی گراف زیر را به دست آورید.



گراف G



ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی



dfs: 0,1,3,7,4,5,2,6

در واقع با شروع از گره 0 تا عمق پیش می رویم و پس از رسیدن به گره 4 چون گره 1 قبل از آن پیمايش شده است

یک قدم به عقب بر می گردیم و گره های 5 ، 2 و 6 را پیمايش می کنیم .

13-7. جست و جوی عرضی (bfs)

برای آشنایی با جست و جوی عرضی ، دو گراف شکل زیر را به این روش جست و جو (پیمايش) می کنیم و سپس

به الگوریتم آن می پردازیم . برای جست و جوی عرضی گراف شکل الف که در واقع یک درخت است ، ابتدا گره

ریش هو سپس هر یک از فرزندان آن مثلا از چپ به راست ملاقات می شوند :

A , B , C , D

سپس فرزندان گره های سطح دوم ملاقات می شوند :

A , B , C , D , E , F , G

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

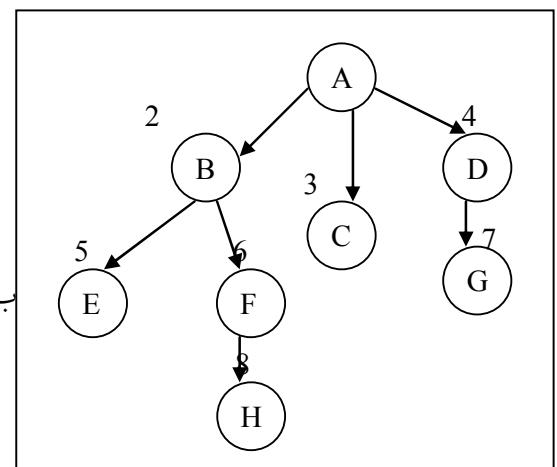
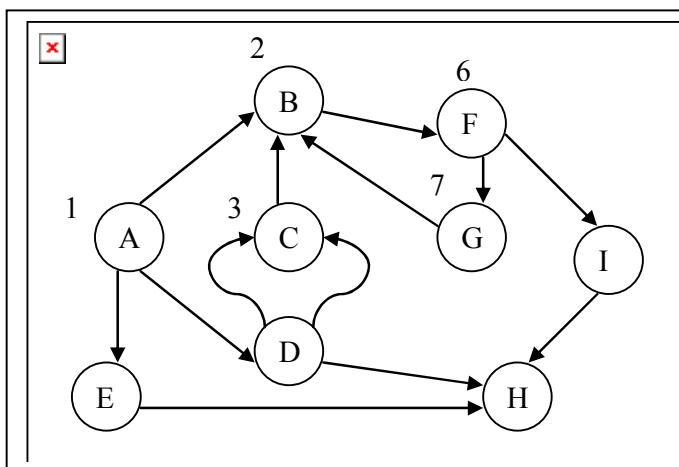
سپس فرزندان گره های سطح سوم ملاقات می شوند :

A , B , C , D , E . F . G , H

در این روش ، گره ها را سطح به سطح پیمایش می کنیم و هنگام ملاقات هر گره در یک سطح ، باید آن را ذخیره کنیم و پس از کامل شدن آن سطح ، دوباره به آن گره برگردیم تا گره های همچوار آن ملاقات شوند. چون اولین گره ای که در سطح فعلی ملاقات شد ، باید اولین گره ای باشد که در برگشت دستیابی می شود ، صفت ساختمان داده مناسبی برای این کار است .

اکنون گراف شکل ب را به روش عرضی ملاقات می کنیم . با شروع از گره A و سپس گره های همچوار A را ملاقات

A , B , D , E می کنیم :



ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

این کار را برای گره D که همچوar بعدی آن است ، تکرار می کنیم و تمام گره های همچوar آن را که ملاقات نشده اند ملاقات می کنیم :

A , B , D , E , F , C , H

سپس این کار را برای E که همچوar بعدی A است تکرار می کنیم و تمام گره های همچوar آن را که ملاقات نشده اند ، ملاقات می کنیم . H تنها گره همچوar E است که قبلاً ملاقات نشده است .

اکنون که گره های همچوar A ، B و D را ملاقات کردیم ، این کار را برای گره های همچوar آن ها که ملاقات نشده اند ، تکرار می کنیم : ابتدا F سپس C و سپس H . گره های G و I همچوar F هستند و آن ها را ملاقات می کنیم :

A , B , D , E , F , C , H , G , I

گره های B، A همچوar C هستند که قبلاً ملاقات شده اند و H گره های زیر ندارد چون همه ی گره ها ملاقات شدند ، جست و جو خاتمه می یابد .

در جست و جوی عرضی نیز همانند جست و جوی عمقی ممکن است نتوانیم با شروع از یک گره خاص ، همه گره ها را ملاقات می کنیم . به عنوان مثال ، جست و جوی عرضی گراف شکل زیر که با B شروع می شود ، گره های A و E را ملاقات نمی کند ، ولی گره های زیر ملاقات می شوند :

E , F , G , I , H , C , D

در این صورت ، الگوریتم باید طوری اصلاح شود که پیمایش را از گره ملاقات نشده دیگری ادامه دهد . این گره را می توان از لیست پیوندی گره ها انتخاب کرد .

الگوریتم جست و جوی عرضی (پیمایش عرضی) bfs

1. فیلد نشانگر تمام گره ها را برابر با 1 قرار دهید ، به معنای آماده پیمایش است .

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

2. گره شروع (مثل A) را در صفت قرار دهید و فیلد وضعیت آن را به 2 تغییر دهید که به معنای انتظار ملاقات است.

3. مراحل 4 و 5 را آنقدر ادامه دهید تا صفت خالی شود.

4. گره ابتدار صفت (مثل N) را حذف کنید. آن را پردازش کنید و وضعیت آن را به حالت پردازش شده 3 تغییر دهید.

5. تمام گره های همجاوار N را که در حالت آماده هستند، در انتهای صفت قرار دهید و وضعیت آن ها را به 2 تغییر دهید.

6. پایان الگوریتم.

توجه کنید که جست و جوی عرضی (bfs) با یک رأس مانند V شروع می شود و پس از ملاقاتات آن را علامت گذاری می کنیم و چنانچه همه رئوس ملاقاتات نشده باشند، رئوس مجاور با اولین رأس مجاور V را ملاقاتات می کنیم. برای پیاده سازی این الگوریتم از یک صفت استفاده می شود که به صورت غیر بازگشتی می توان الگوریتم

مورد بحث را به فرم زیر بیان نمود:

Procedure bfs(V : integer) ;

Begin

Visited [V] := true;

Addq(q,V) ;

While not (is empty (q) do

Begin

Delq (q,v);

For all node w adjacent to v do

Begin

Addq(q,v);

Visited [w] := true ;

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

End ;

End ;

End ;

14-7. الگوریتم وارشال برای محاسبه ماتریس مسیر

ماتریس path_k را طوری تعریف می کنیم که $[\text{i}, \text{j}]$ path_k وقتی درست باشد که مسیری از گره i به گره j وجود داشته باشد که از گره هایی با شماره بزرگتر از k (احتمالاً به جر خود i و j) عبور نکند. مقدار $[\text{i}, \text{j}]_{k+1}$ path_{k+1} چگونه از ماتریس path_k بدست می آید؟ بدیهی است که برای هر i و j که $[\text{i}, \text{j}]_{k+1}$ ارزش درستی دارد (برابر با true است)، $[\text{i}, \text{j}]_{k+1}$ نیز باید ارزش درستی داشته باشد (چرا؟). تنها حالتی که با نادرست بودن path_k ، $[\text{i}, \text{j}]_{k+1}$ عبور از گره i به گره j وجود داشته باشد که مسیری از i به j وجود داشته باشد که از گره $k+1$ عبور کند، ولی مسیری از i به j وجود نداشته باشد که فقط از گره های ۱ تا k عبور کند. معناش این است که باید مسیری از گره i به گره j وجود داشته باشد که فقط از گره های ۱ تا k تا گره j وجود دارد. بنابراین، $[\text{i}, \text{j}]_{k+1}$ در یکی از دو حالت زیر ارزش درستی دارد:

$\text{Path}_k[\text{i}, \text{j}] = \text{true}$

$\text{Path}_{k+1}[\text{i}, \text{k}+1] = \text{true}$ and $\text{Path}_{k+1}[\text{k}+1, \text{j}] = \text{true}$

از این دو حالت نتیجه می گیریم که $[\text{i}, \text{j}]_{k+1}$ path_{k+1} برابر است با:

$\text{Path}_k[\text{i}, \text{j}] \text{ or } (\text{Path}_k[\text{i}, \text{k}+1] \text{ and } \text{Path}_{k+1}[\text{k}+1, \text{j}])$

بنابراین، الگوریتم به دست آوردن ماتریس path_k از ماتریس path_{k-1} به صورت زیر است (n تعداد گره ها است):

الگوریتم محاسبه ماتریس path_k از path_{k-1}

For $i := 1$ to n do

 For $j := 1$ to n do

$\text{Path}_k[\text{i}, \text{j}] := \text{path}_{k-1}[\text{i}, \text{k}] \text{ or } (\text{path}_{k-1}[\text{i}, \text{k}] \text{ and } \text{path}_{k-1}[\text{k}, \text{j}]);$

این الگوریتم می تواند به صورت زیر خلاصه شود تا کارایی آن افزایش یابد:

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

For $i := 1$ to n do

 For $j := 1$ to n do

$\text{Path}_k[i,j] := \text{path}_{k-1}[i,j];$

 For $I := 0$ to n do

 If $\text{path}_{k-1}[i,k] = \text{true}$ then

 For $j=0$ to n do

$\text{Path}_k[i,j] := \text{path}_{k-1}[i,j] \text{ or } \text{path}_{k-1}[k,j];$

بدیهی است که $\text{path}[i,j] = T$ است که ماتریس همجواری است. زیرا تنها راه رسیدن از گره i به گره j

بدون عبور از هر گره دیگر، رفتن مستقیم از i به j است. علاوه بر این $\text{path}_{n-1}[i,j] = \text{path}[i,j]$ است

، زیرا اگر مسیری از گره i با شماره 0 تا $n-1$ عبور کند، می تواند هر مسیری از گره i به j را انتخاب نماید. بنابراین ،

رویه زیر می تواند ماتریس مسیر (بستار انتقالی گراف) را محاسبه کند (این روش توسط وارشال ارائه شد و الگوریتم

وارشال نام دارد). کارایی این الگوریتم $O(n^2)$ است که n تعداد گره های گراف است.

7-15. گراف وزن دار

گاهی به یال های گراف وزن داده می شود و به این گراف ، **گراف وزن دار** یا شبکه گویند. این وزن می تواند

فاصله یک گره تا گره دیگر ، یا هزینه رفتن از یک گره به گره مجاور باشد . بنابراین در هنگام نمایش گراف وزن دار با

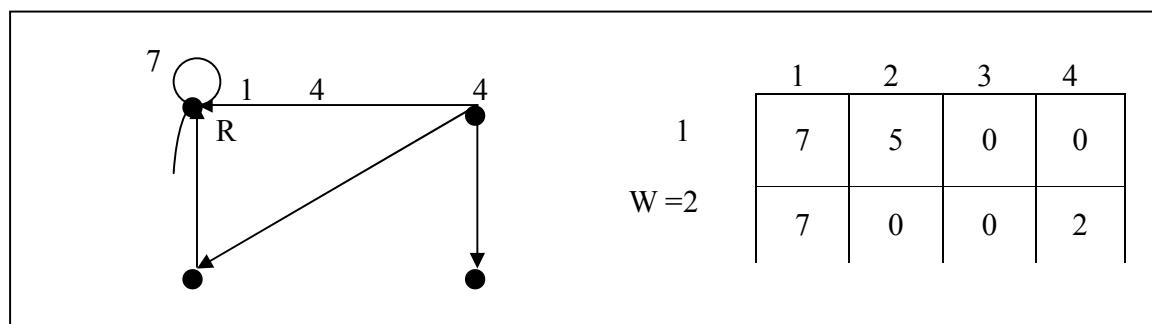
استفاده از ماتریس همجواری ، باید این وزن ها نیز نگهداری شوند. در این صورت ماتریس همجواری را **ماتریس**

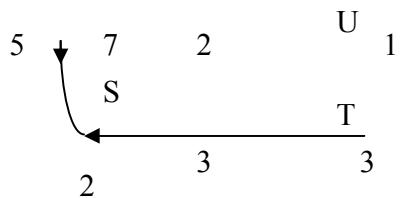
وزن می گویند و آن را با W نشان می دهیم . در ماتریس وزن ، وقتی بین دو گره i و j یال وجود دارد ، در عنصر

$[i,j]$ به جای مقدار 1 ، وزن آن یال قرار می گیرد . به عنوان مثال ، در شکل زیر یک گراف وزن دار و ماتریس وزن

آن را مشاهده می کنید. نمونه ای از گراف وزن دار می تواند شهر های یک کشور و راه های بین آن ها باشد که وزن

هر یال ، فاصله بین دو شهر یا میانگین زمان لازم برای طی مسافت بین دو شهر است .





گراف وزن دار

ماتریس همچو اری (ماتریس وزن)

16-7. الگوریتم گوتاهترین مسیر

ممکن است گراف برای نمایش شاهراه های یک استان یا کشور به کار گرفته شود ، به طوری که رأس های گراف نماینده شهرها و یال های آن بینگر جاده ها ی بین آن ها باشد . می توان به یال های این گراف وزن داد . وزن هر یال می تواند فاصله بین دو شهر یا میانگین زمان لازم برای پیمودن فاصله بین دو شهر باشد . فرض کنید شخصی می خواهد از شهر A به شهر B برود . این شخص با سؤال های زیر روبرو است :

1- آیا فقط یک مسیر از A به B وجود دارد ؟

2- اگر بیش از یک مسیر بین A و B وجود داشته باشد ، کوتاه ترین مسیر کدام است ؟

در حل این مسئله ها می توان طول مسیر را به حای تعداد یال ها ، مجموع وزن های آن ها در نظر گرفت . چون خیابان ها و جاده ها می توانند یک طرفه باشند برای حل این مسئله از گراف جمعت دار استفاده مس کنیم . در نمایش ماتریس وزن گراف ، اگر مسیری بین دو گره وجود نداشته باشد ، عدد خیلی بزرگی در نظر می گیریم ، به طوری که از وزن هر یالی بیشتر باشد . معناش این است که هزینه طی مسیر بین آن دو گره بسیار زیاد است . برای تعیین کوتاه ترین مسیر ، از

دو الگوریتم می توان استفاده کرد :

1. الگوریتم وارشال

2. الگوریتم دیکسترا

16-7. الگوریتم وارشال برای تعیین ماتریس کوتاه ترین مسیر

یک گراف جهت دار به نام G و با گره های V_1, V_2, \dots, V_n را در نظر بگیرید ، به طوری که وزن هر یال ، مقداری

مثبت است . همان طور که می دانید ، این گراف را می توان با استفاده از ماتریس وزن آن (w) نمایش داد .

ماتریس مسیر path که قبل مشاهده کردید بیانگر این است که آیا مسیرهایی بین گره های مفروض وجود دارد یا خیر .

اکنون می خواهیم ماتریس q را پیدا کنیم که ماتریس کوتاه ترین مسیر نام دارد ، به طوری که q_{ij} طول کوتاهترین

مسیر از گره V_i به V_j است . برای به دست آوردن ماتریس q ، تغییر اندکی در الگوریتم وارشال ایجاد می کنیم ،

به طوری که دنباله ای از ماتریس های q_0, q_1, \dots, q_n را تعریف می کنیم (مثل ماتریس های مسیر_k ، path_{k+1}) . عناصر این ماتریس به صورت زیر تعریف می شوند :

$q_k[i,j] =$ کوچک ترین طول مسیر قبل از V_i به V_j ، یا مجموع طول های مسیرهای قبلی از V_i به V_k و از V_k به V_j

به عبارت دیگر ، داریم :

ماتریس اولیه q_0 برابر با ماتریس وزن است ، با ای تفاوت که در آن ، به جای صفر ، مقدار بزرگی (∞) قرار می گیرد .

آخرین ماتریس به دست آمده ، ماتریس کوتاه ترین مسیر خواهد بود. این الگوریتم را می توان به صورت زیر بیان کرد:

Procedure shortpath($w:ar; var q:ar;$);

Var

$i,j,k:integer;$

$q1:ar;$

Begin

$q := w; \{q \text{ starts off as weight matrix.}\}$

for $k:=1$ to m do

begin

for $i := 1$ to m do

```
for j:=1 to m do  
    q1[i,j] := min(q[i,j],q[i,k],q[k,j]);  
    q := q1;  
end;  
end;
```

16-7. الگوریتم دیکسترا

این الگوریتم می تواند در یک شبکه که نام دیگر آن گراف جهت دار وزن دار است، کوتاه ترین مسیر را از گره ای به گره دیگر پیدا کند. گراف شکل زیر (الف) را در نظر بگیرید. کوتاه ترین مسیر از V_0 به بقیه نقاط در شکل (ب) آمده است. در مورد شکل (ج) در ادامه توضیح می دهیم. اگر مخواهیم کوتاه ترین مسیر از S به t را محاسبه کنیم، S را گره مبدأ و t را گره مقصد می نامیم.

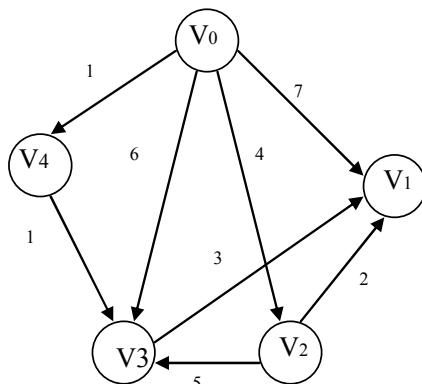
فرض می کنیم، S حاوی مجموعه ای از رئوس است که کوتاه ترین مسیر را دارند. در آغاز، S حاوی مبدأ (V_0) است. برای هر V که در S نمی باشد، $\text{distance}[v]$ طول کوتاه ترین مسیر از V_0 به V است، به طوری که این مسیر از رئوی می گذرد که در S وجود دارند. بر اساس این مفاهیم، یافتن مسیرهایی که به ترتیب طولشان به صورت نزولی باشند، موارد زیر مطرح است:

1. اگر کوتاه ترین مسیر بعدی، به رأس u باشد، آن گاه مسیری که از V_0 آغاز و به u ختم شد، فقط از رئوی می گذرد که در S وجود دارند. فرض کنید رأسی مثل v در این مسیر وجود دارد که در S نیست. در این صورت، مسیر V_0 به u شامل مسیری از V_0 به V است که طول آن کمتر از مسیر V_0 به u می باشد. با این فرض، کوتاه ترین مسیرها به ترتیب غیر نزولی طولشان تولید می شوند و در نتیجه مسیر کوتاه تر V_0 به V قبل تولید شده است. نتیجه می گیریم که هیچ رأسی بین V_0 به u نیست که در S باشد.

2. اگر در بین رئوی مسیری که در S نیستند، مسیری به مقصد u ایجاد شود، این مسیر کمترین فاصله را دارد و با $\text{distance}[u]$ مشخص می شود. این موضوع، از تعریف distance و نکته اول نتیجه می شود. اگر

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

چندین رأس با distance مساوی وجود داشته باشند در S نباشد، هر کدام از آن ها را می توان انتخاب کرد.



(الف)

مبدأ	مقصد	مسیر	طول
V0	V4	V0V4	1
V0	V3	V0V4V3	2
V0	V2	V0V2	4
V0	V1	V0V4V3V1	5

(ب)

	0	1	2	3	4
0	100	7	4	6	1
1	100	100	100	100	100
2	100	2	100	5	100
3	100	3	100	100	100
4	100	100	100	1	100

(ج)

3. با انتخاب رأس u و تولید کوتاه ترین مسیر از $V0$ به u ، رأس u در S ذخیره می شود. یعنی u به عنوان عضوی از S در نظر گرفته می شود. با اضافه شدن به S ، ممکن است فاصله کوتاه ترین مسیرهایی که از $V0$ شروع می شود و فقط از رأس های موجود در S می گذرد و به رأسی مثل V ختم می شود که در S نیست، تغییر کند. اگر این فاصله تغییر کند، آن گاه باید مسیر کوتاه تری از $V0$ تا V را بیابیم. این مسیر از u می گذرد. رئوس میانی موجود در این مسیر در S است و می توان مسیرهایی از u به V را انتخاب کرد که هیچ رأس مسانی در آن ها نباشد. طول کوتاه ترین مسیر برابر با $W[u,v]$ است که W ماتریس وزن است.

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

الگوریتم دیکسترا با استفاده از این مشاهدات می توان طول کوتاه ترین مسیره از V_0 به هر رأس دیگر در گراف را بیابد . برای پیاده سازی الگوریتم ، مجموعه S به صورت آرایه ای مثل found نگهداری می شود . اگر رأس i در W نباشد ، $\text{found}[i]$ برابر با $false$ و گرنه برابر با $true$ است . خود گراف با ماتریس همچواری به نام W مشخص شده است $[j,i] \in W$ وزن یال $< j,i >$ است . اگر بین رأس i و رأس j مسیری وجود نداشته باشد ، $W[i,j]$ برابر با عدد خیلی بزرگ خواهد شد . این عدد باید دو شرط زیر را داشته باشد :

1. این عدد باید از هر یک از مقادیر وزن یال ها بیشتر باشد .

2. این عدد باید طوری انتخاب شود که $\text{distance}[u] + W[u,v]$ موجب سرریز نشود .

براساس آن چه که گفته شد ، ماتریس همچواری گراف شکل (الف) به صورت شکل (ج) نشان داده شده است در این شکل ، برای سهولت ، وزن یال بین رأس هایی که مسیری بین آن ها وجود ندارد 100 انتخاب شده است .

یعنی 100 بیانگر این است که بین دو رأس هیچ مسیری وجود ندارد .

الگوریتم دیکسترا برای یافتن کوتاه ترین مسیر

مرحله مقدماتی : آرایه distance ، ماتریس W و مجموعه های S ، Y و N را به صورت زیر تعریف می کیم :

$$W \text{ ماتریس وزن} \quad \bullet$$

$$S \text{ مجموعه ای از گره هاست که کوتاه ترین مسیر آن ها مشخص شد} \quad \bullet$$

$$N \text{ مجموعه تمام رأس ها} \quad \bullet$$

$$Y = N - X \quad \bullet$$

را برابر با صفر قرار می دهیم .

مرحله اصلی :

$$d_q = d_p + c_{pq} = \min \{d_i + w_{ij}\} \quad i \in S_j \subset Y$$

و قرار می دهیم :

$$S = q \cup X$$

اگر مرحله اصلی الگوریتم را $n-1$ بار اجرا کنیم ، کوتاه ترین مسیر از گره به گره $n-1$ به دست می آید .

اکنون این الگوریتم را بر روی گراف شکل (الف) اجرا می کنیم :

مرحله اول :

	0	1	2	3	4
Distance	0		4	2	1
S	0				

$$Y = \{1,2,3,4\}$$

$$\begin{aligned} \text{Min } \{d_0 + w_{01}, d_0 + w_{02}, d_0 + c_{04}\} &= \min \{0+7, 0+4, 0+6, 0+1\} \\ &= \min \{7, 4, 6, 1\} = 1 \end{aligned}$$

مقدار 1 ناشی از $d_0 + w_{04}$ است . در نتیجه رأس 4 را به S اضافه کرده ، فاصله آن را در آرایه distance برابر با 1 تعیین می کنیم .

	0	1	2	3	4
Distance	0				1
S	0				

$$S = \{0,4\}$$

$$Y = \{1,2,3\}$$

مرحله دوم :

$$\begin{aligned} \text{Min } \{d_0 + w_{01}, d_0 + w_{02}, d_0 + w_{03}, d_0 + w_{43}\} &= \min \{0+7, 0+4, 0+6, 1+1\} \\ &= \min \{7, 4, 6, 2\} = 2 \end{aligned}$$

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

مقدار 2 ناشی از $d_{04} + w_{43}$ است . در نتیجه رأس 3 را به S اضافه کرده ، فاصله آن را در آرایه distance برابر با 2

تعیین می کنیم .

	0	1	2	3	4
Distance	0			2	1

$$S = \{0, 4, 3\}$$

$$Y = \{1, 2\}$$

مرحله سوم :

(از 3 به 2 مسیری وجود ندارد) .

$$\text{Min } \{d_0 + w_{01}, d_0 + w_{02}, d_3 + w_{31}\} = \min \{0+7, 0+4, 2+3\}$$

$$= \min \{7, 4, 5\} = 4$$

مقدار 4 ناشی از $d_0 + w_{02}$ است . در نتیجه رأس 2 را به S اضافه کرده ، فاصله آن را در آرایه distance برابر با 5

تعیین می کنیم .

	0	1	2	3	4
Distance	0		4	2	1

$$S = \{0, 4, 3, 2\}$$

$$Y = \{1\}$$

مرحله چهارم :

$$\text{Min } \{d_0 + w_{01}, d_2 + w_{21}, d_3 + w_{31}\} = \min \{0+7, 4+2, 2+3\}$$

ساختمان داده ها - آموزشکده فنی محمودآباد/مدرس : جواد وحیدی

$$= \min\{7, 6, 5\} = 5$$

مقدار 5 ناشی از $w_{31} + d_3$ است . در نتیجه رأس 1 را به S اضافه کرده ، فاصله آن را در آرایه distance برابر با 5 تعیین می کنیم .

	0	1	2	3	4
Distance	0	5	4	2	1

$$S = \{0, 1, 2, 3, 4\}$$

$$Y = \{\}$$

هر وقت رأس مقصد (t) در S قرار گرفت ، d_t کوتاه ترین مسیر از مبدأ به مقصد است . بنابراین ، کوتاه ترین مسیر از V0 به V1 برابر با 5 است . در واقع آرایه distance کوتاه ترین مسیر از مبدأ (V0) به هر گره دیگر را نشان می دهد . بنابراین ، با توجه به آرایه distance ، کوتاه ترین مسیرهای به دست آمده از v0 به هر گره دیگر ، همان چیزی است که در شکل (ب) مشاهده می کنید .

7-17. درخت حداقل پوشای (فراگیر مینیمال)

زیرگرافی از یک گراف که از کلیه رئوس آن بگذرد و درخت نیز باشد ، درخت فراگیر نامیده می شود . درخت حداقل پوشای ، یک درخت فراگیر است که مجموع وزن های متناظر با یال های آن مینیمم باشد . الگوریتم زیر که به الگوریتم کروسکال مشهور است ، برای تعیین درخت حداقل پوشای کار می رود .

الگوریتم کروسکال

i=0 . قرار دهید

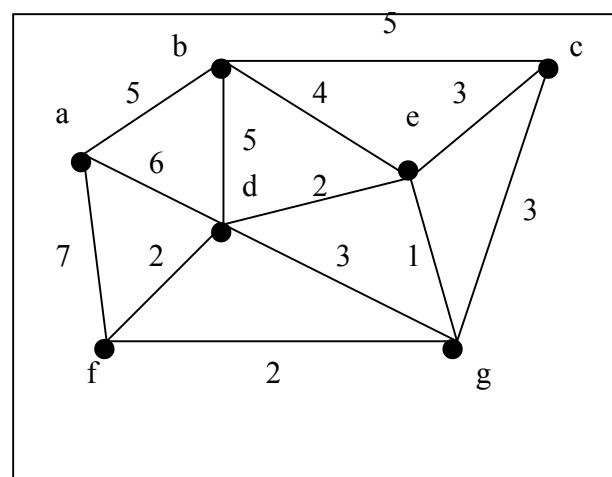
1. یال e_1 را چنان اختیار کنید که وزن متناظر ب آن مینیمم باشد .

2. به ازای $i \leq n-2$ ، اگر یال های e_1, e_2, \dots, e_i انتخاب شده باشند ، آن گاه یال e_{i+1} را از بین یال های باقیمانده چنان اختیار می کنیم که وزن متناظر با آن حتی الامکان کوچک باشد و دور نیز ایجاد نشود .

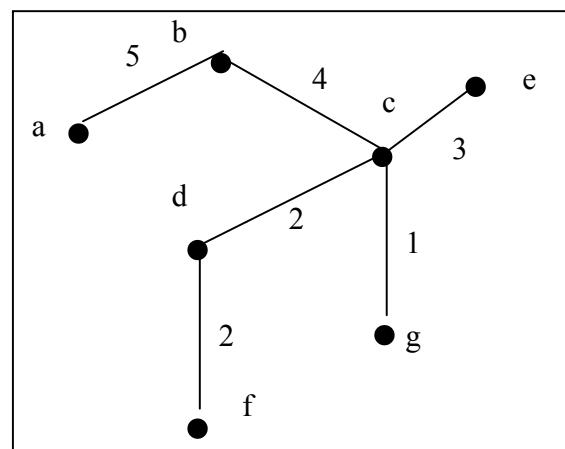
روند را تا دستیابی به یک درخت حداقل پوشایی دهیم .

مثال

درخت حداقل پوشای (فراگیر مینیمال) گراف زیر را باید :



حل : با استفاده از الگوریتم کروستال و با شروع از یال $\{e,g\}$ که دارای وزن 1 است ، درخت زیر حاصل می شود :



وزن متناظر با این درخت برابر است با :

$$1 + 2 + 2 + 4 + 5 = 17$$

ساختمان داده ها- آموزشکده فنی محمودآباد/مدرس : جواد وحیدی
