به نام خدا

ساختمان های گسسته - نظریه گراف

Discrete Structures – Graph Theory

مدرس : محمد امین زاده

www.sooraac.ir

نظریه گراف

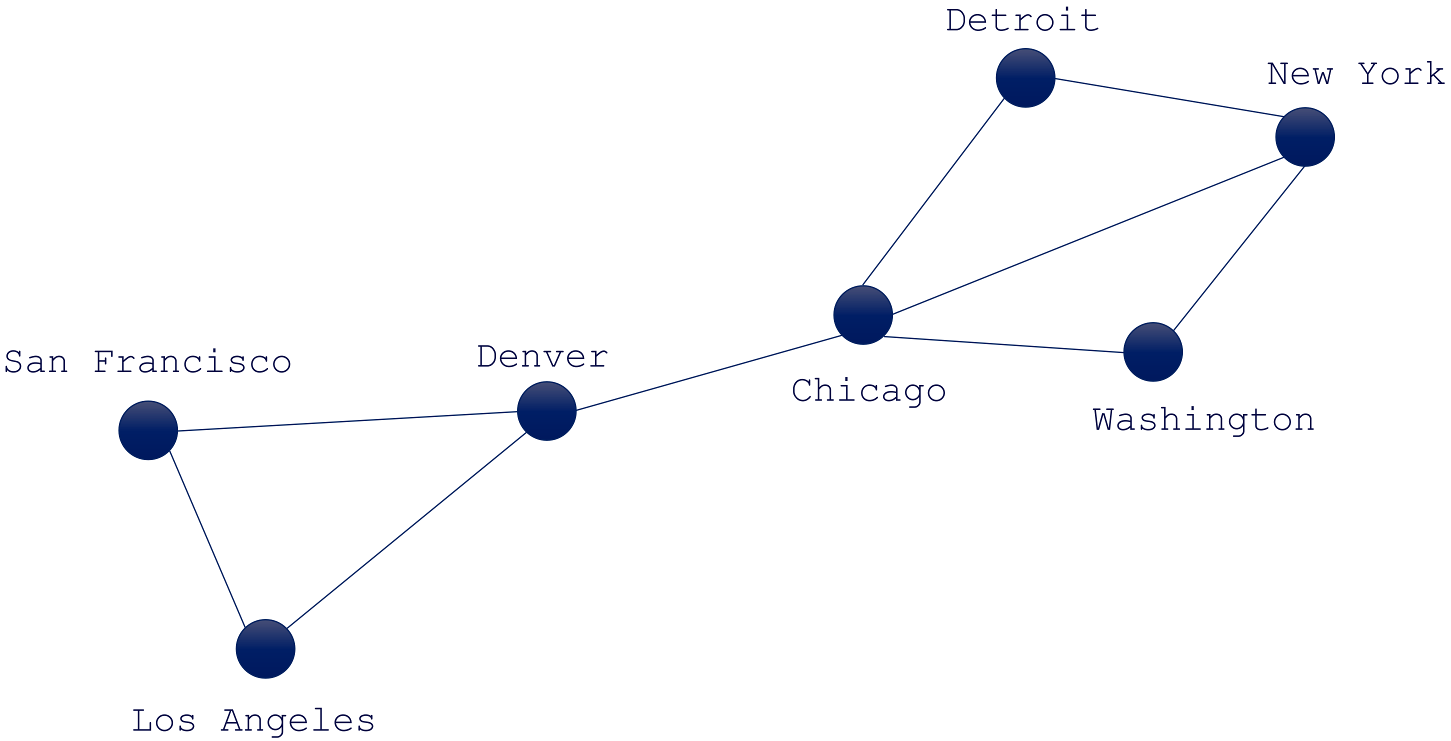
❖ **تعریف ۱. گراف (ساده)** $(\boldsymbol{SimpleGraph})$ $G=(V,E)$

شامل یک مجموعه غیر تهی از رئوس $(Vertices)$ **به نام** V **و یک مجموعه یال ها** $(Edges)$ **به نام** E **می باشد.**

❖ **تعریف ۲.** هر یال با دو راس در ارتباط است که نقاط انتهایی $(EndPoints)$ آن **یال گفته می شوند. مجموعه یال یک مجموعه زوج مرتب از رئوس است.**

- در گراف غیرجهت دار ترتیب زوج ها مهم نیست.

❖ تمرین ۱. یک گراف که ارتباط بین ایالت ها را مشخص می کند.



❖ تمرین ۱. یک گراف که ارتباط بین ایالت ها را مشخص می کند.

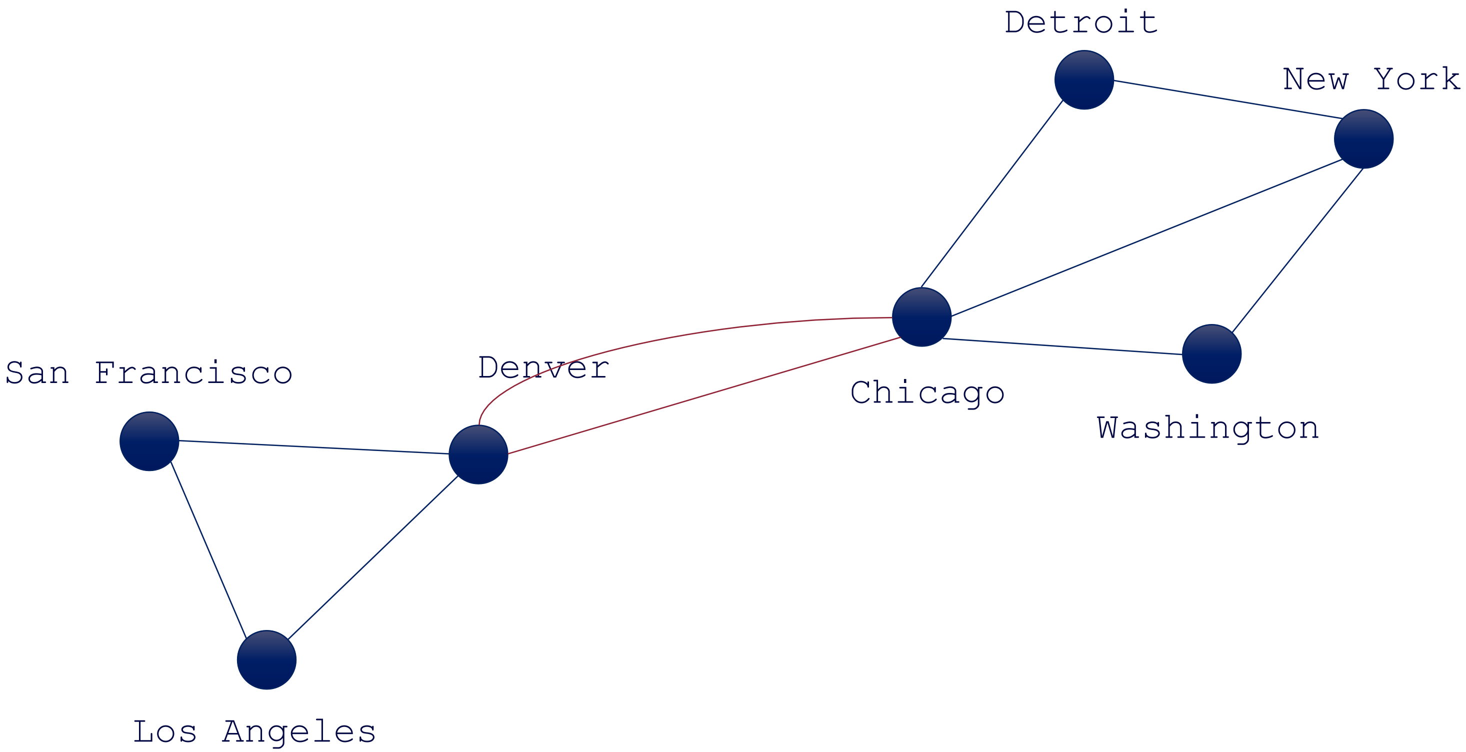
- مجموعه رئوس :

```
V = { Chicago, Denver, Detroit, Los Angeles,
    New York, San Francisco, Washington }
```

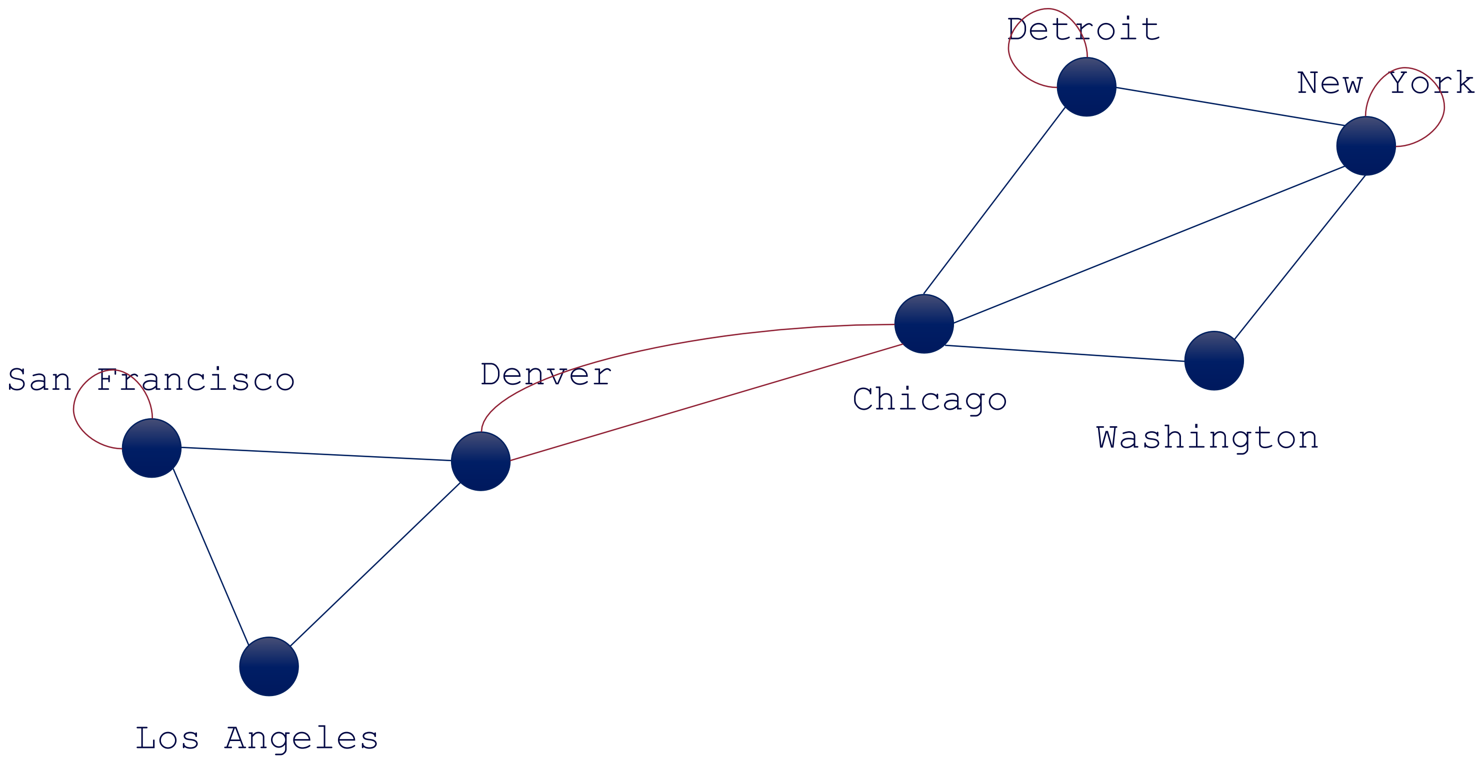
- مجموعه یال ها :

```
E = { (San Francisco, Los Angeles),(San Francisco, Denver),
      (Los Angeles, Denver),(Denver, Chicago),
      (Chicago, Detroit),(Detroit, New York),
      (New York, Washington),(Chicago, Washington),
      (Chicago, New York) }
```

❖ **تعریف ۳. مالتی گراف** $(MultiGraph)$ گرافی می گویند که بین رئوس بیش از یک یال قرار گیرد.

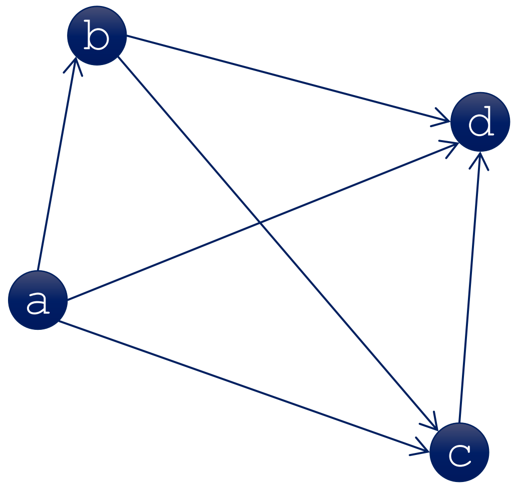


❖ تعریف ۴. سودو گراف به گرافی می گویند که هم دارای یال چندگانه $(MultiEdge)$ باشد و هم طوقه $(Loop)$ بتواند در آن باشد.



❖ تعریف ۵. گراف جهتدار $(\boldsymbol{DirectedGraph})$ $G=(V,E)$

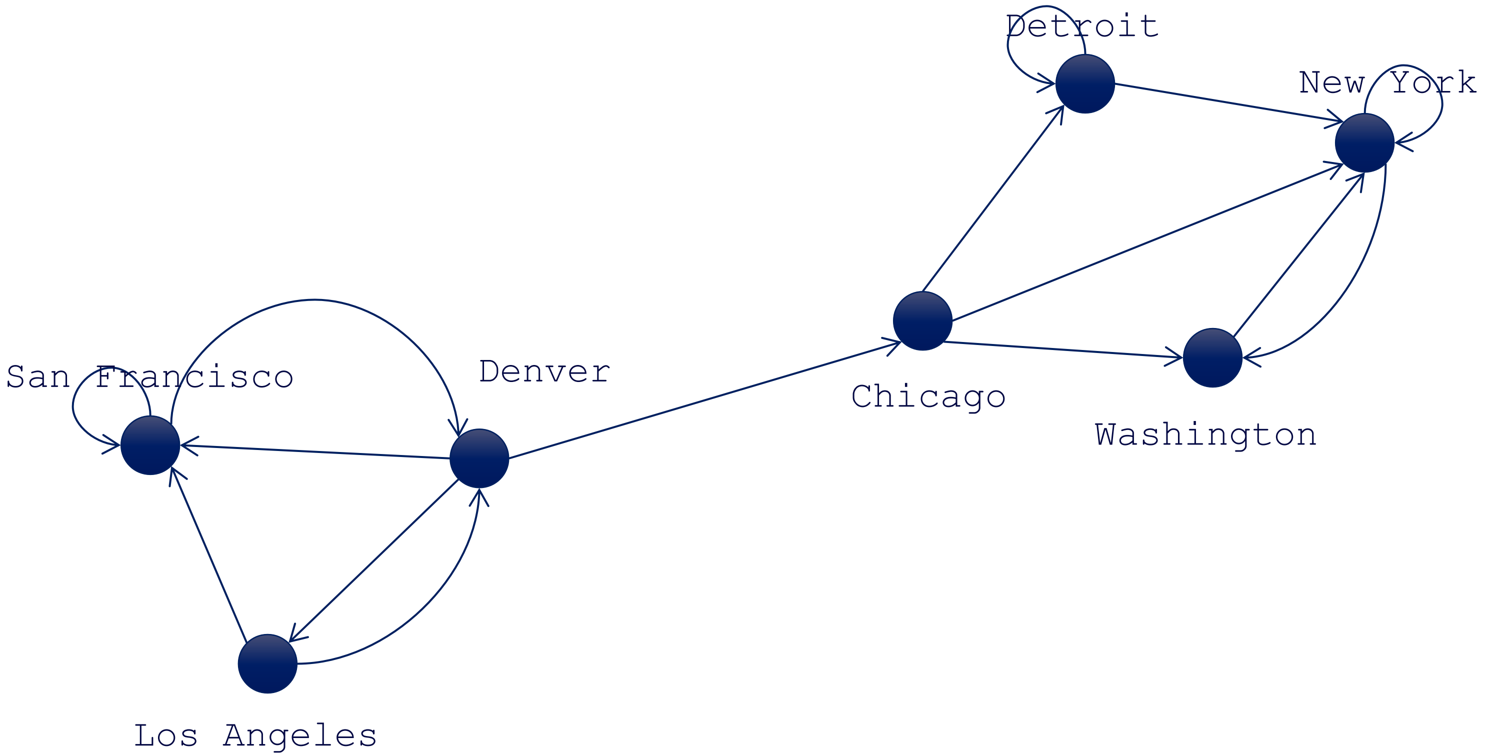
شامل یک مجموعه غیر تهی از رئوس $(Vertices)$ به نام V و یک مجموعه یال ها $(Edges)$ به نام E می باشد.



- مجموعه یال یک مجموعه زوج مرتب از رئوس است که در آن ترتیب زوج ها مشخص کننده دو یال جداگانه است.

- به یال در گراف جهت دار بردار (Arc) نیز می گویند.

❖ **تعریف ۶. مالتی گراف جهتدار** $(DirectedMultiGraph)$ گرافی جهتدار می گویند که هم دارای یال چندگانه $(MultiEdge)$ باشد و هم طوقه $(Loop)$ بتواند در آن باشد.



❖ **تعریف** ۷. دو راس را که با یک یال به هم وصل هستند را **راس مجاور** $(Adjacent)$ **یا راس همسایه** $(Neighbor)$ می نامیم.

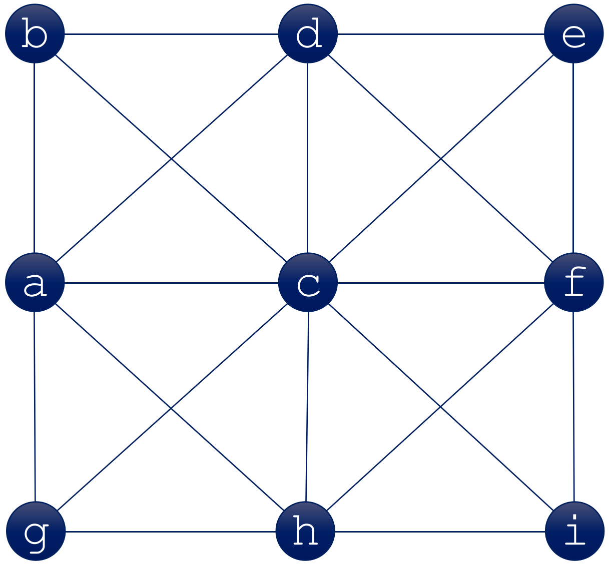
❖ **تعریف** ۸. **درجه هر راس برابر تعداد یال های متصل به آن راس است.**

- درجه راس v را با $deg(v)$ نمایش می دهیم.
- برای سودو گراف ، حلقه به تنهایی ۲ تا به درجه آن راس اضافه می کند.

❖ تمرین ۲. در گراف زیر درجه هر راس و مجموع درجات را تعیین کنید.

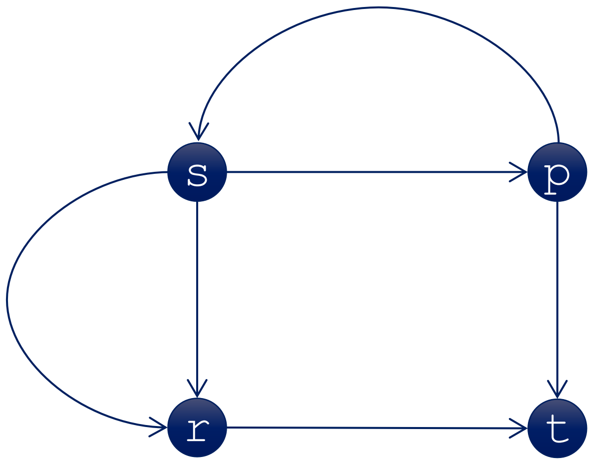
V	$deg(V)$
a	5
b	3
c	8
d	5
e	3
f	5
g	3
h	5
i	3

▪ جدول ۱ - درجه رئوس



➢ مجموع درجات برابر با ۴۰ می باشد.

❖ **تمرین ۳. در یک مسابقه تیم s یکبار تیم p ، تیم r یکبار تیم t ، تیم s دوبار تیم r ، تیم p یکبار تیم t و تیم p یکبار تیم s را می برد ، با استفاده از گراف ، این مسابقه را مدلسازی کنید.**



- **قضیه ۱. قضیه دست دادن** $(HandShaking\ Theorem)$

فرض کنید $G = (V,E)$ یک **گراف بی سو** باشد و E تعداد یال های گراف G باشد.

$$2E = \sum_{v \in V} deg(v)$$

- **قضیه ۲.** اگر یک **گراف بی سو** باشد **تعداد یال با درجه فرد** ، **زوج** است.

❖ **تعریف ۹.** وقتی (u,v) **یک یال جهت دارگراف** G **باشد می گوییم** u **مجاور به** v $(adjacent\ to)$ **است و یا** v **مجاور از** $(adjacent\ from)$ u **است. به** u **راس اولیه** $(initial)$ **و به** v **راس پایانی** (end) **می گویند.**

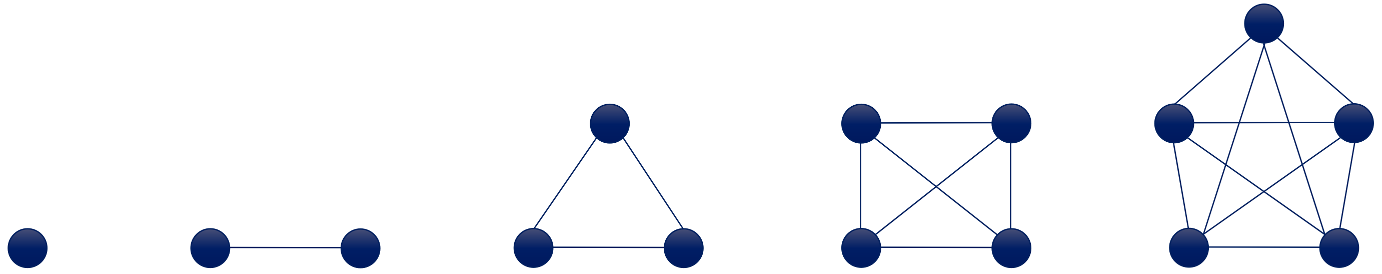
❖ **تعریف ۱۰.** در گراف جهتدار راس v **درجه ورودی** $(in-degree)$ بیانگر تعداد **بردار ورودی** به آن راس است و درجه ورودی راس v را با $deg^{-}(v)$ نمایش می دهیم. به طور مشابه راس v **درجه خروجی** $(out-degree)$ بیانگر تعداد **بردار خروجی** از آن راس است و درجه خروجی راس v را با $deg^{+}(v)$ نمایش می دهیم .

❖ **قضیه ۳. فرض کنید $G = (V,E)$ یک گراف جهتدار باشد و E تعداد یال های گراف G باشد.**

$$\sum_{v\in V} deg^{-}(v) = \sum_{v\in V} deg^{+}(v) = |E|$$

❖ **تعریف ۱۱. گراف کامل** ($Complete$) ، **گرافی است که در آن بین هر دو راس حتما یک یال وجود دارد.**

➢ گراف کامل با n راس را با K_n نمایش می دهیم.



K_1 K_2 K_3 K_4 K_5

- گراف کامل K_n برای $1 \le n \le 5$

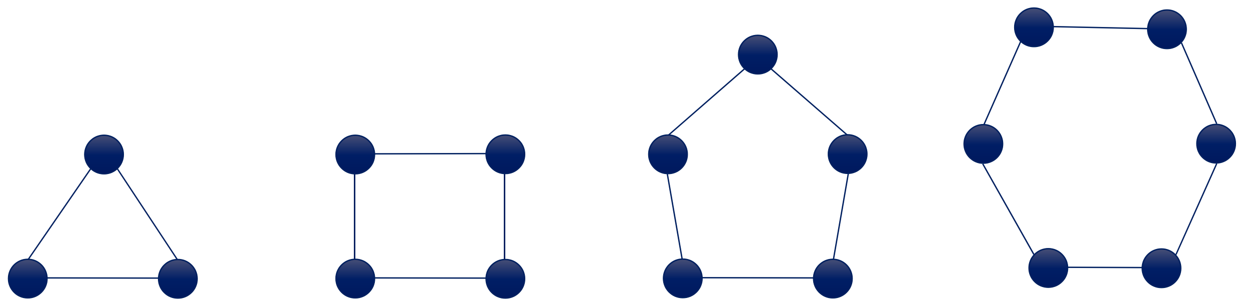
❖ تمرین ۴. ثابت کنید در گراف کامل K_n تعداد یال ها برابر با $\binom{n}{2}$ است.

در گراف کامل درجه هر راس برابر با $n-1$ می باشد. بنابراین با توجه به اینکه n راس داریم مجموع درجه کل رئوس برابر با $n(n-1)$ می باشد.

$$2E = \sum_{v \in V} deg(v) = n(n-1) \quad \Rightarrow \quad E = \frac{n(n-1)}{2} = \binom{n}{2}$$

❖ **تعریف ۱۲. گراف دور** $(Cycle)$ ، گرافی است که تنها یک دور کامل به طول n (اندازه تعداد رئوس گراف) دارد که از همه رئوس آن عبور می کند.

➢ گراف دور با n راس را با C_n نمایش می دهیم.

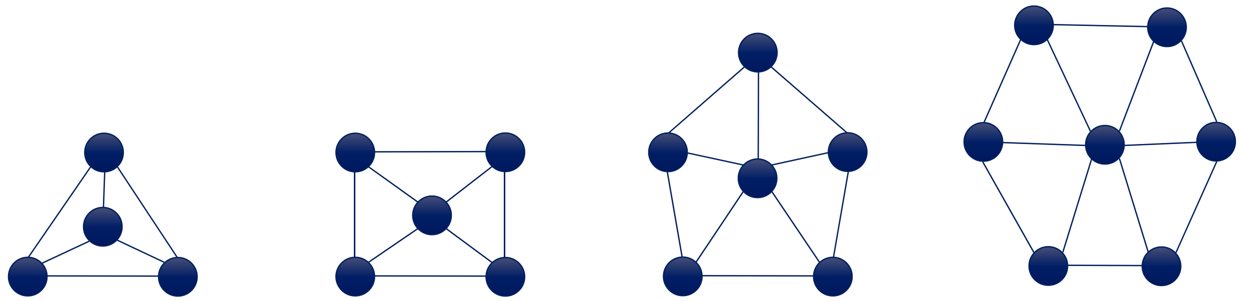


C_3 C_4 C_5 C_6

▪ گراف دور C_n برای $3 \le n \le 6$

❖ **تعریف ۱۳. گراف چرخ** $(Wheel)$ ، گرافی است که اگر به یک گراف دور C_n یک راس جدید اضافه کنیم و از هر راس گراف C_n یک یال به آن متصل نماییم تشکیل می شود.

➢ گراف چرخ با n راس را با W_n نمایش می دهیم.

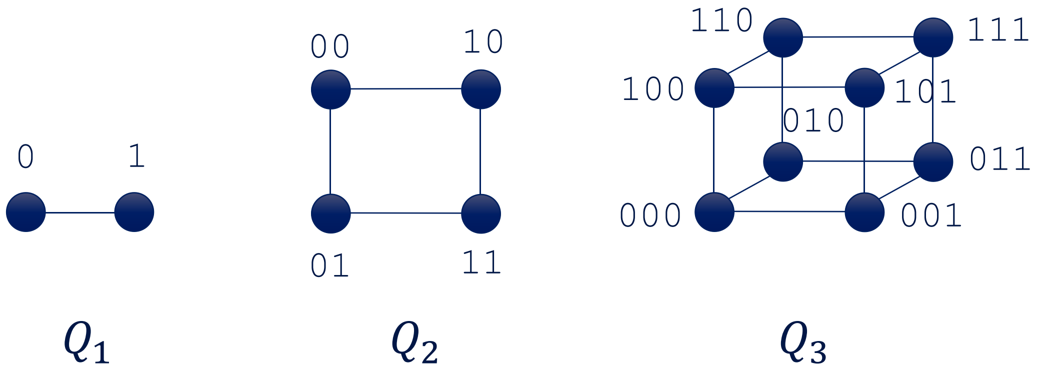


K_3 K_4 K_5 K_6

▪ گراف دور W_n برای $3 \leq n \leq 6$

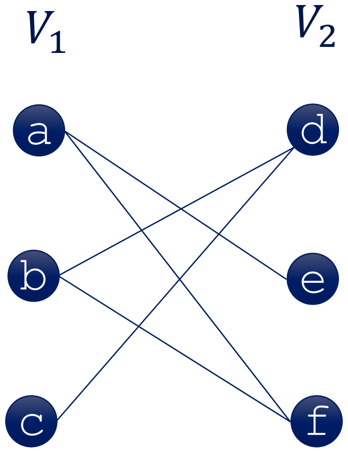
❖ **تعریف ۱۴. گراف n-مکعب** ، گرافی است که رئوس آن را با رشته صفر و یک به طول n نامگذاری شده و بین هر دو راس که تنها در **یک بیت** با یکدیگر **تفاوت** دارند یک یال رسم شده است.

➢ گراف **n-مکعب** با 2^n راس را با Q_n نمایش می دهیم.

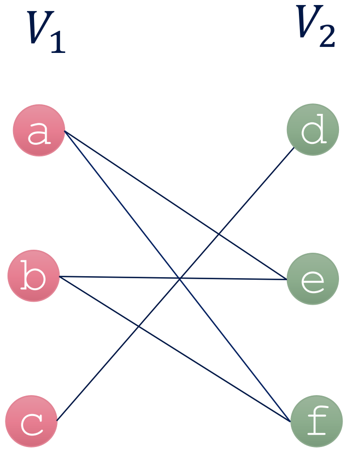


▪ گراف n-مکعب Q_n برای $1 \leq n \leq 3$

❖ **تعریف ۱۵.** یک گراف ساده $G(V,E)$ را یک **گراف دوبخشی** $(\boldsymbol{Bipartite})$ می نامند وقتی بتوان مجموعه راس های گراف را به مجموعه مجزای V_1 و V_2 افراز نمود به صورتی که هر یال، یک راس از یکی از مجموعه ها را به راسی در مجموعه دیگر متصل نماید.

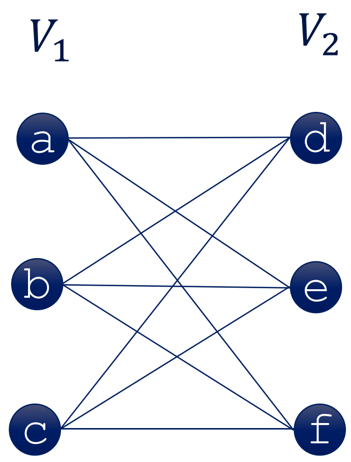


❖ **قضیه ۴**. یک گراف ساده **دوبخشی** است اگر و فقط اگر بتوان با دو تنها دو رنگ راس های گراف را رنگ نمود به طوری که هیچ دو راس مجاوری از یک رنگ نباشند.

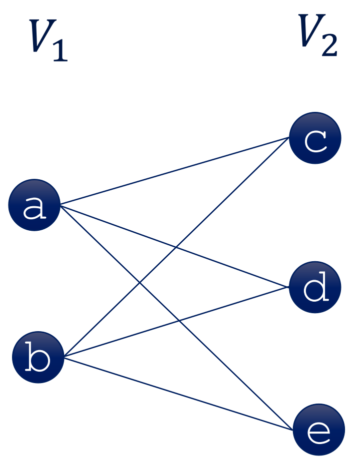


❖ **تعریف ۱۶. یک گراف ساده $G(V,E)$ را یک گراف دوبخشی کامل** ($\boldsymbol{Complete\ Bipartite}$) می نامند اگر هر راس از یک قسمت به تمام رئوس از قسمت دیگر مجاور باشد.

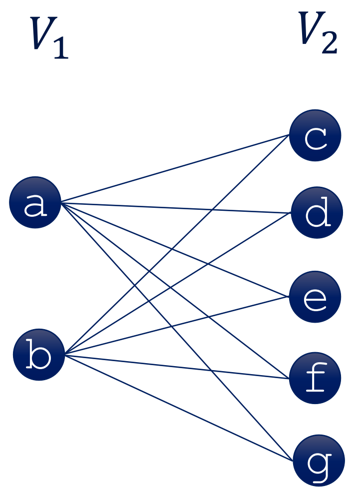
➢ گراف دو بخشی کامل را با $K_{n,m}$ نمایش می دهیم (n,m تعداد رئوس دو قسمت هستند) .



$K_{3,3}$

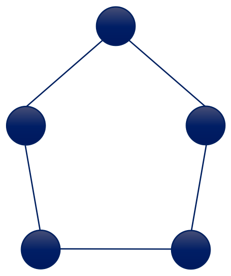


$K_{2,3}$

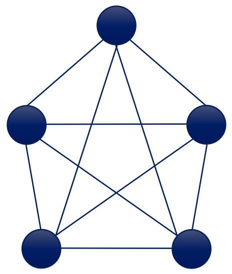


$K_{2,5}$

❖ **تعریف ۱۷.** یک **زیرگراف** از گراف $G=(V,E)$ را به صورت گراف زیر تعریف می کنیم : $H=(W,F)$ اگر $W \subseteq V$ و $F \subseteq E$



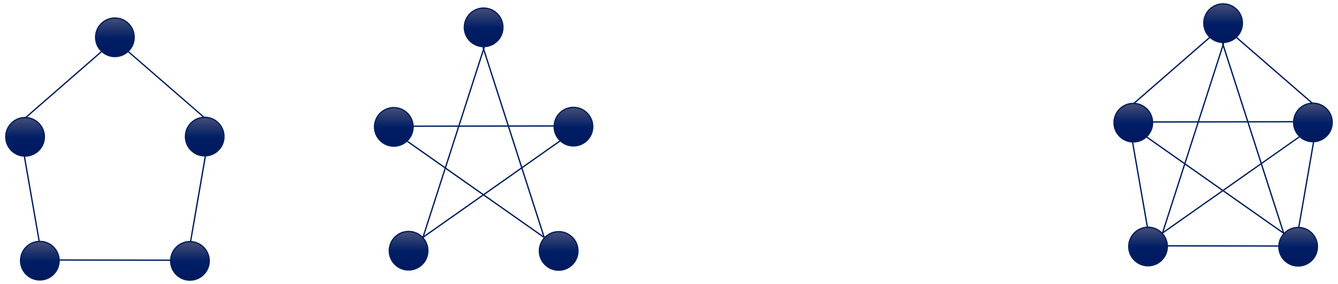
C_5



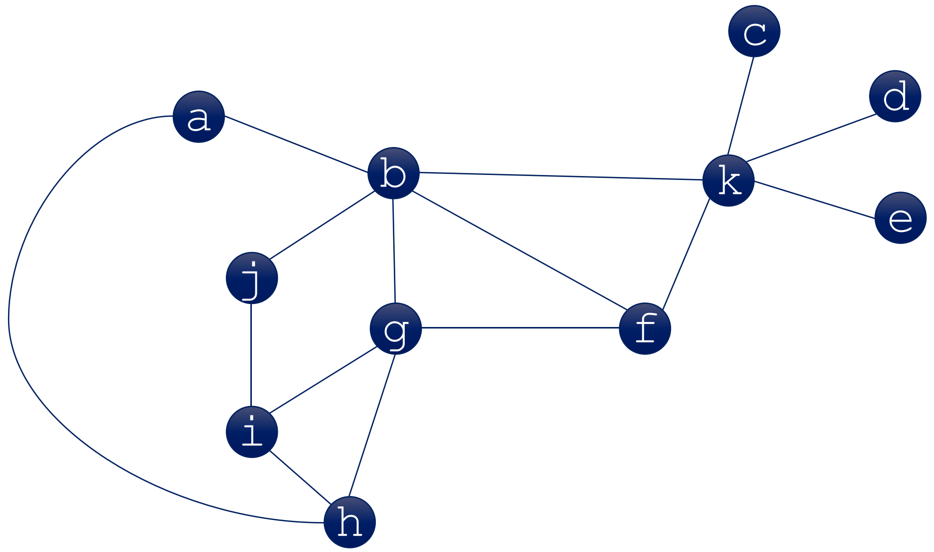
K_5

❖ تعریف ۱۸. اجتماع دو گراف $(Union)$ ساده $G_1(V_1,E_1)$ و $G_2(V_2,E_2)$ را به صورت زیر تعریف می کنیم :

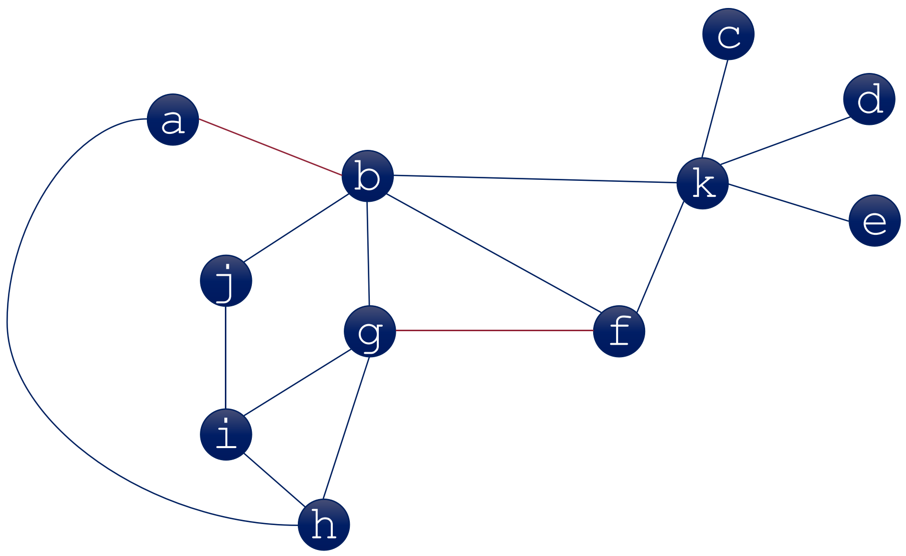
$$G\ (V,E)\ =\ G_1(V_1,E_1)\ \boldsymbol{U}\ G_2(V_2,E_2) =\ G\ (V_1\ \boldsymbol{U}\ V_2\ ,\ E_1\ \boldsymbol{U}\ E_2)$$



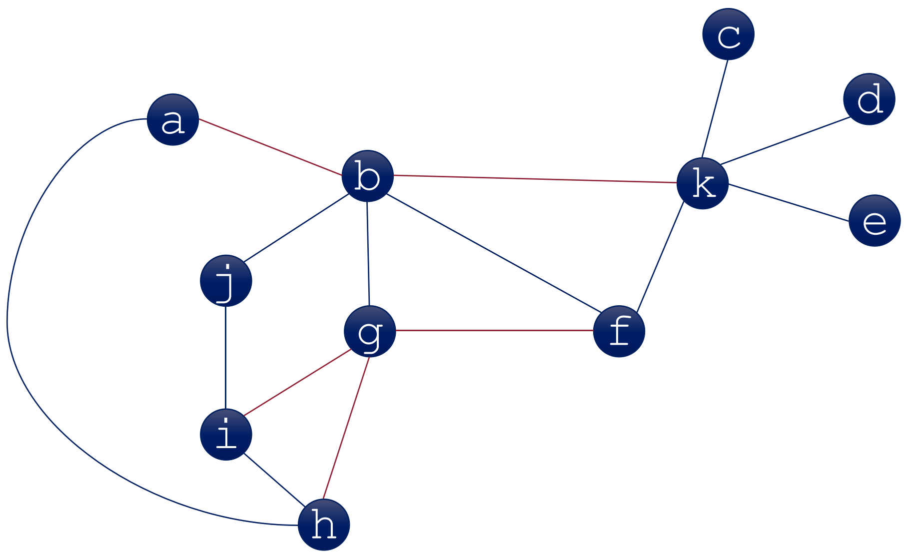
❖ تمرین ۵. در گراف زیر راس ها ، اداره را نمایش می دهند. یک یال ، دو اداره را به هم وصل می کند مشروط بر اینکه یک ارتباط مخابراتی بین آن دو اداره وجود داشته باشد هر اداره می تواند با اداره دیگری از طریق یک ارتباط مستقیم مخابراتی یا از طریق اداره دیگری که پیغام را تقویت می کند ارتباط مخابراتی داشته باشد.



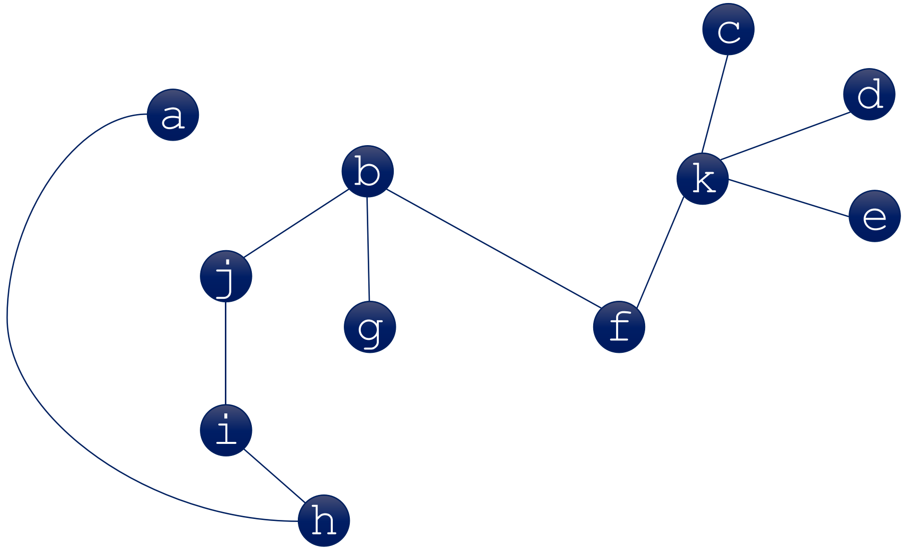
❖ تمرین ۵. الف) با ارائه یک مثال نشان دهید ارتباط مخابراتی بین تمام ادارات حتی اگر بعضی از خطوط ارتباطی قطع شود ، نیز برقرار می شود.



❖ تمرین ۵. ب) حداکثر خطوطی که می تواند بین ادارات قطع شود اما ارتباط همچنان برقرار باشد ، چند تا است؟



❖ تمرین ۵. ج) گرافی رسم کنید حداکثر خطوطی که می تواند بین ادارات قطع شود اما ارتباط همچنان برقرار باشد را نشان دهد.



الگوریتم دایجسترا

Dijkstra's Algorithm

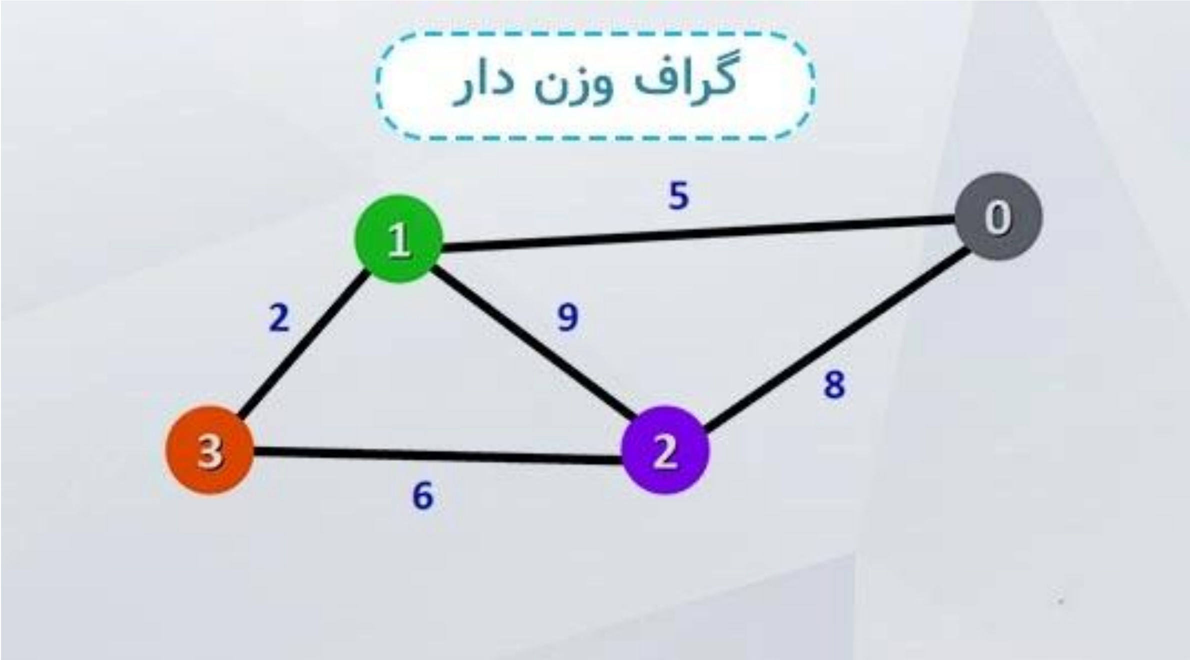
- الگوریتم دایجسترا (Dijkstra's Algorithm) یکی از مهم‌ترین الگوریتم‌های نظریه گراف است که برای یافتن کوتاه‌ترین مسیر استفاده می‌شود. این الگوریتم در فناوری‌های امروزی کاربردهای گسترده‌ای دارد، از جمله مسیریابی در نقشه‌ها و شبکه‌های کامپیوتری. در این ارائه، ابتدا مروری کوتاه بر مفاهیم گراف و پیش‌نیازهای لازم خواهیم داشت و سپس به بررسی نحوه عملکرد این الگوریتم می‌پردازیم.

- گراف‌ها می‌توانند جهت‌دار یا غیرجهت‌دار باشند.

- گراف غیر جهت دار: اگر در هر جفت گره که به هم متصل هستند، بتوان از یک نود به نود دیگر در هر دو جهت رفت، آن گراف غیرجهت دار است.
- گراف جهت دار: اگر به ازای هر جفت نود متصل، تنها بتوان از یک نود به نود دیگر در یک جهت مشخص رفت، آن گراف جهت دار است. در این مدل از گراف ما از فلش به جای خطوط ساده برای نمایش جهت یال استفاده می‌کنیم.

❖ گراف وزن دار

گراف وزن دار، گرافی است که یال‌های آن دارای وزن یا هزینه هستند. وزن هر یال می‌تواند نشان دهنده‌ی فاصله، زمان یا هر چیزی که در آن مدل، دو نود را به هم متصل می‌کند باشد. به عنوان مثال در گراف وزن دار زیر شماره‌های با رنگ آبی روی هر یال نمایشگر وزن آن یال است.

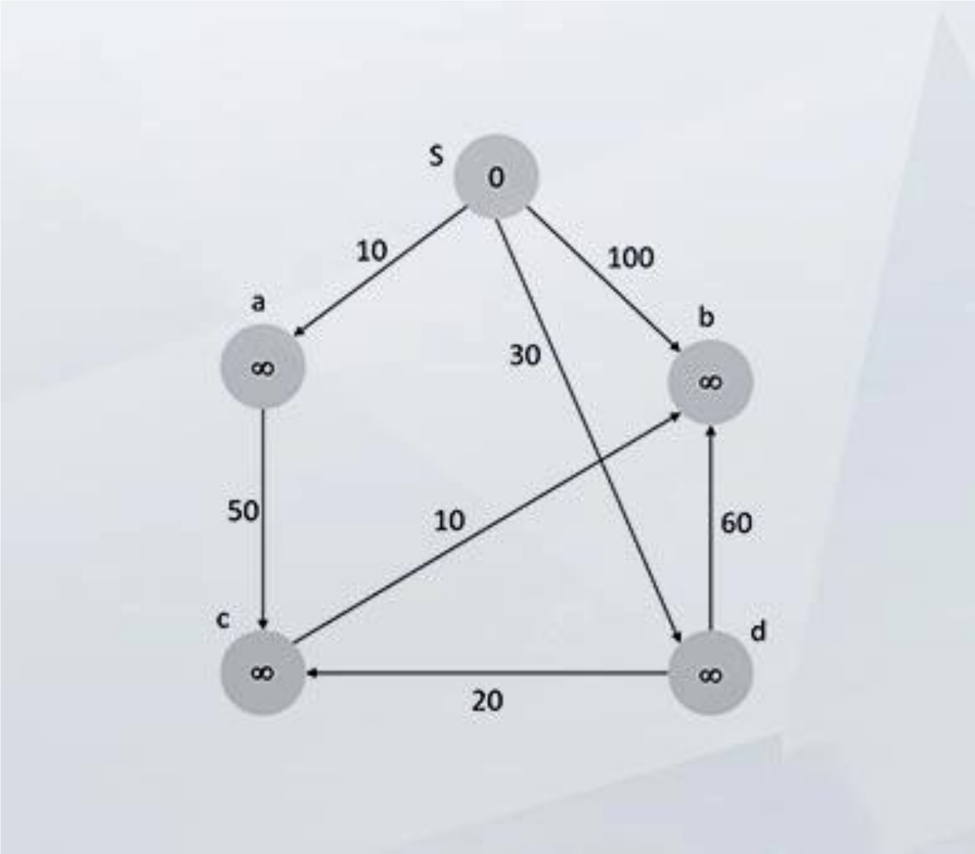


❖ الگوریتم دایجسترا Dijkstra’s Algorithm

❖ پس از آشنایی با مفاهیم پایه‌ای گراف، به بررسی الگوریتم دایجسترا می‌پردازیم که یکی از مهم‌ترین الگوریتم‌ها در نظریه گراف است. این الگوریتم توسط «ادسگر دایکسترا»، دانشمند برجسته علوم کامپیوتر، در سال ۱۹۵۹ معرفی شد. او در مقاله‌ای با عنوان A note on two problems in connexion with graphs این الگوریتم را ارائه داد و پایه‌ای برای حل مسائل مسیریابی در گراف‌ها ایجاد کرد.

❖ الگوریتم دایجسترا در گراف، یک روش حریصانه برای یافتن کوتاه ترین مسیر از یک مبداء واحد اتخاذ می‌کند. در این الگوریتم لازم است که وزن تمامی یال‌ها غیر منفی باشد چرا که در غیر این صورت الگوریتم قادر به حل مسأله نیست.

❖ با فرض این که مثال مورد نظر یافتن کوتاه ترین مسیر از گره مبداء s در گراف G=(V,E) باشد الگوریتم دایجسترا با شروع از رأس s کوتاه ترین مسیرها را از مبداء s به تمامی رئوس دیگر پیدا می‌کند. برای درک بهتر مراحل اجرای این الگوریتم به مثال زیر توجه کنید:



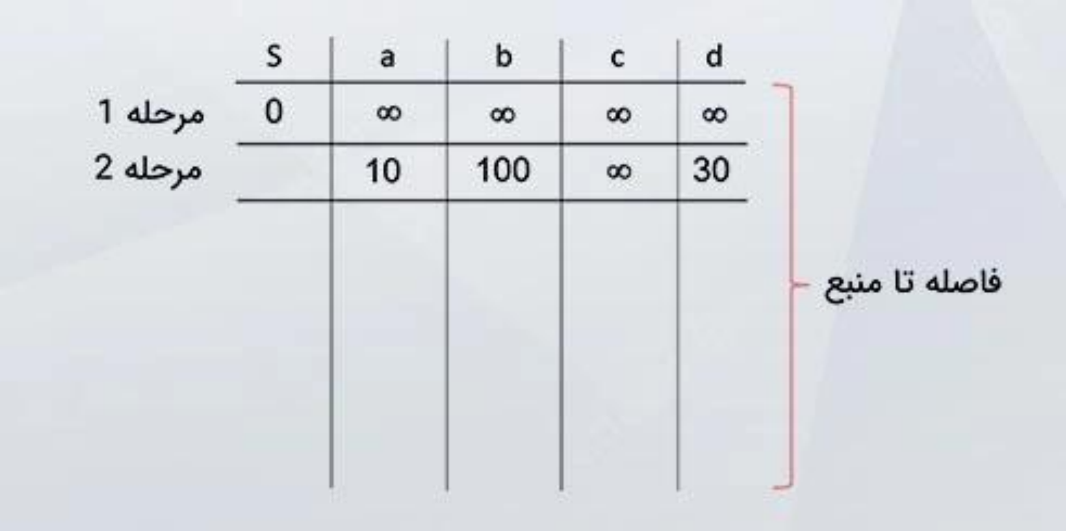
همان‌طور که از شکل مشخص است ابتدا تمامی رئوس به غیر از منبع s با ∞ علامت‌گذاری شده‌اند. یعنی هنوز مسیری از s به آن‌ها کشف نشده است. در مرحله بعد کوتاه‌ترین مسیر بین رأس s و تمامی رئوسی که از s مستقیما یالی به آن‌ها وجود دارد محاسبه می‌شود.

رأس بعدی که انتخاب می‌شود همان رأسی است که کوتاه‌ترین مسیر را به s داشته (انتخاب حریصانه) و مراحل ذکر شده برای رأس منبع اکنون برای این رأس تکرار شده و کوتاه‌ترین مسیرها از رأس فعلی به تمامی رئوسی که از رأس مذکور مستقیما یالی وجود دارد محاسبه می‌شود. لیستی به صورت زیر که فاصله گره‌ها تا منبع S را نمایش می‌دهد تهیه می‌کنیم.

	S	a	b	c	d
فاصله تا منبع	0	∞	∞	∞	∞

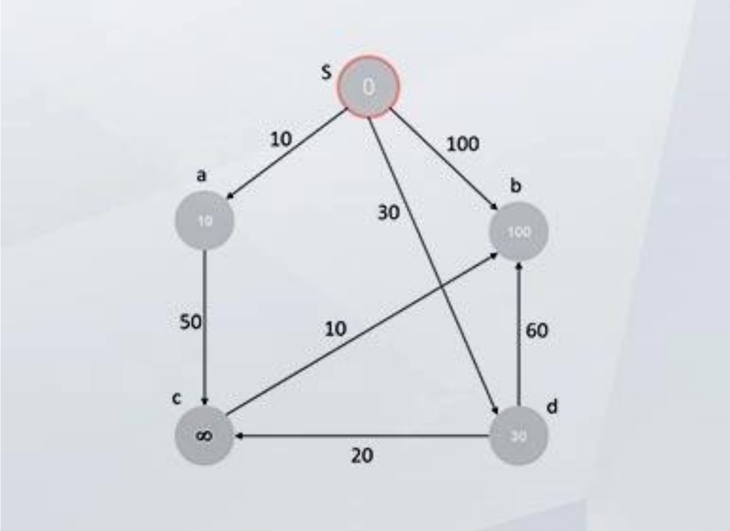
در مرحله اول گره sرا انتخاب و کوتاه‌ترین فاصله از این گره تا هر یک از گره‌های مجاور آن را محاسبه می‌کنیم و سپس بعد از محاسبه، گره s را از لیست حذف می‌کنیم. به عنوان مثال در ابتدا فاصله s تا گره a برابر بینهایت است.

در گراف داده شده با عبور از مسیر با وزن ۱۰ می‌توان از گره s به گره a رسید و از آنجایی که ۱۰ کوچکتر از ∞ (فاصله فعلی گره sتا گره aاست) پس کوتاه‌ترین فاصله از گره s تا گره a در لیست فوق، بروز می‌شود. این امر برای گره‌های b و d نیز که مجاور با s هستند رخ می‌دهد. در نهایت پس از پایان این مرحله به لیست زیر می‌رسیم:



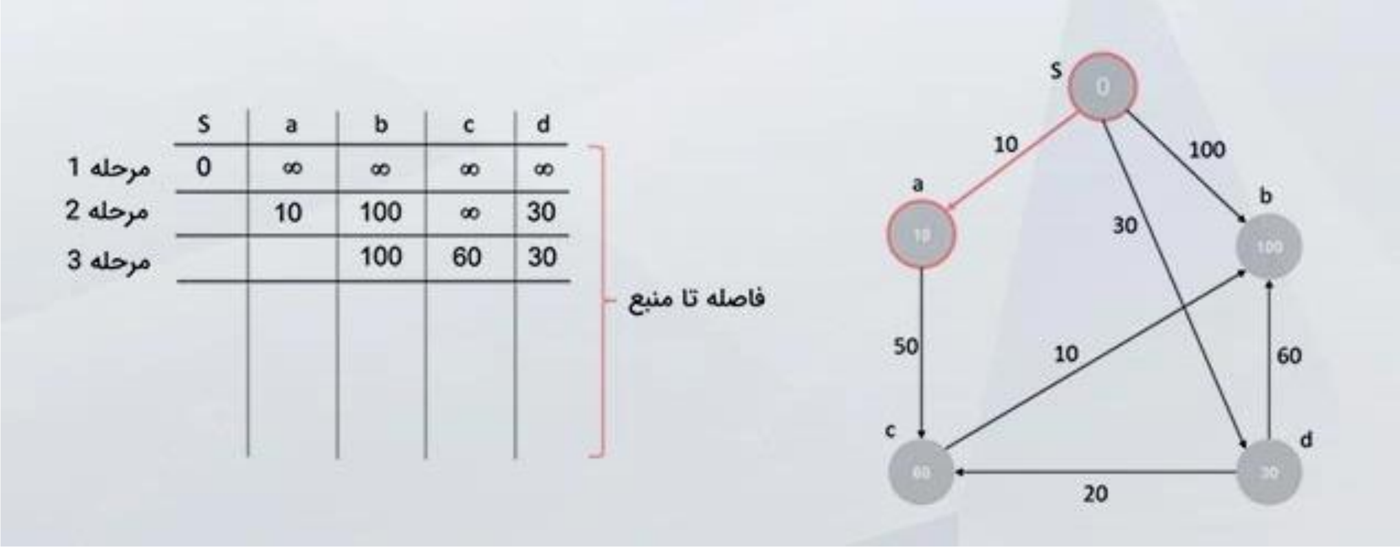
	S	a	b	c	d
مرحله 1	0	∞	∞	∞	∞
مرحله 2		10	100	∞	30

فاصله تا منبع



حال در لیست جدید بدست آمده (مرحله ۲) گره با فاصله min، که در اینجا گره a است، را انتخاب کرده و مراحل فوق را روی آن انجام می‌دهیم. یعنی گره a را انتخاب و کوتاه‌ترین فاصله از این گره تا هر یک از گره‌های مجاور آن را محاسبه می‌کنیم و سپس بعد از محاسبه، گره a را از لیست حذف می‌کنیم.

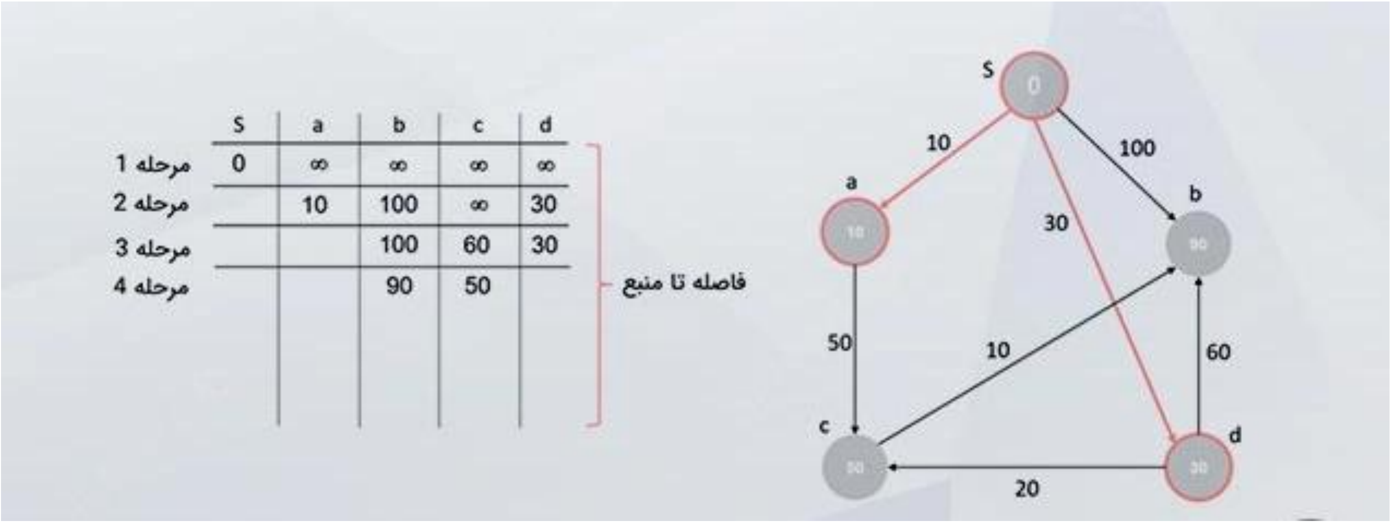
تنها گره مجاور a نود c است که با هزینه ۵۰ می‌توان به آن رفت. همچنین بر اساس لیست، فاصله گره منبع تا c برابر بینهایت است. با گذر از یال با وزن ۵۰ که بین دو گره a و c است، می‌توان با مجموع ۶۰ (۱۰ + ۵۰) از گره s به گره a و سپس به گره c رسید. از آنجایی که ۶۰ از ∞ کوچکتر است میبایست لیست و گراف داده شده را بروز کرد.



حال در لیست جدید بدست آمده (مرحله ۳) گره با فاصله min، که در اینجا گره d است، را انتخاب کرده و مراحل فوق را روی آن انجام می‌دهیم. یعنی گره d را انتخاب و کوتاه‌ترین فاصله از این گره تا هر یک از گره‌های مجاور آن را محاسبه می‌کنیم و سپس بعد از محاسبه، گره d را از لیست حذف می‌کنیم.

گره‌های مجاور d، c و b هستند. از گره منبع تا گره d به اندازه ۳۰ تا هزینه دارد. حال اگر بخواهیم در ادامه‌اش از گره d به گره b برویم با توجه به اینکه هزینه d تا b برابر ۶۰ است، در نتیجه هزینه مسیر s->d->b برابر ۳۰ + ۶۰ یعنی ۹۰ است. این هزینه کمتر از هزینه مسیر فعلی‌اش یعنی مسیر s->b است. در نتیجه مسیر با هزینه کمتر را در لیست و گراف‌مان بروز می‌کنیم.

حال اگر بخواهیم بعد از اینکه از گره منبع به گره dآمدیم، از dبه cبرویم، با توجه به اینکه هزینه d تا c برابر ۲۰ است در نتیجه هزینه مسیر s->d->c برابر ۳۰ + ۲۰ یعنی ۵۰ است. این هزینه کمتر از هزینه مسیر فعلی‌اش یعنی مسیر s->a->cاست. در نتیجه مسیر با هزینه کمتر را در لیست و گراف‌مان بروز می‌کنیم. به شکل زیر توجه کنید.



	s	a	b	c	d
مرحله 1	0	∞	∞	∞	∞
مرحله 2		10	100	∞	30
مرحله 3			100	60	30
مرحله 4			90	50	

حال در لیست جدید بدست آمده (مرحله ۴) گره با فاصله min، که در اینجا گره c است، را انتخاب کرده و مراحل فوق را روی آن انجام می‌دهیم. یعنی گره c را انتخاب و کوتاه‌ترین فاصله از این گره تا هر یک از گره‌های مجاور آن را محاسبه می‌کنیم و سپس بعد از محاسبه، گره c را از لیست حذف می‌کنیم.

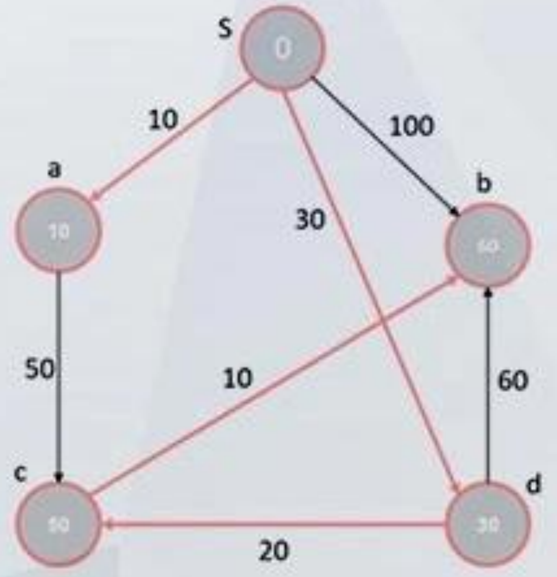
تنها گره مجاور c نود b است که با هزینه ۱۰ می‌توان به آن رفت. حال با توجه به اینکه کوتاه‌ترین مسیر از منبع تا نود c برابر ۵۰ است (بر اساس آخرین بروز رسانی لیست) در نتیجه می‌توان با رفتن از گره c به b، یعنی با هزینه ۵۰ + ۱۰ = ۶۰، از گره منبع به گره b رفت. s->d->c->b

از آنجایی که ۶۰ از ۹۰ کوچکتر است میبایست لیست و گراف داده شده را بروز کرد.

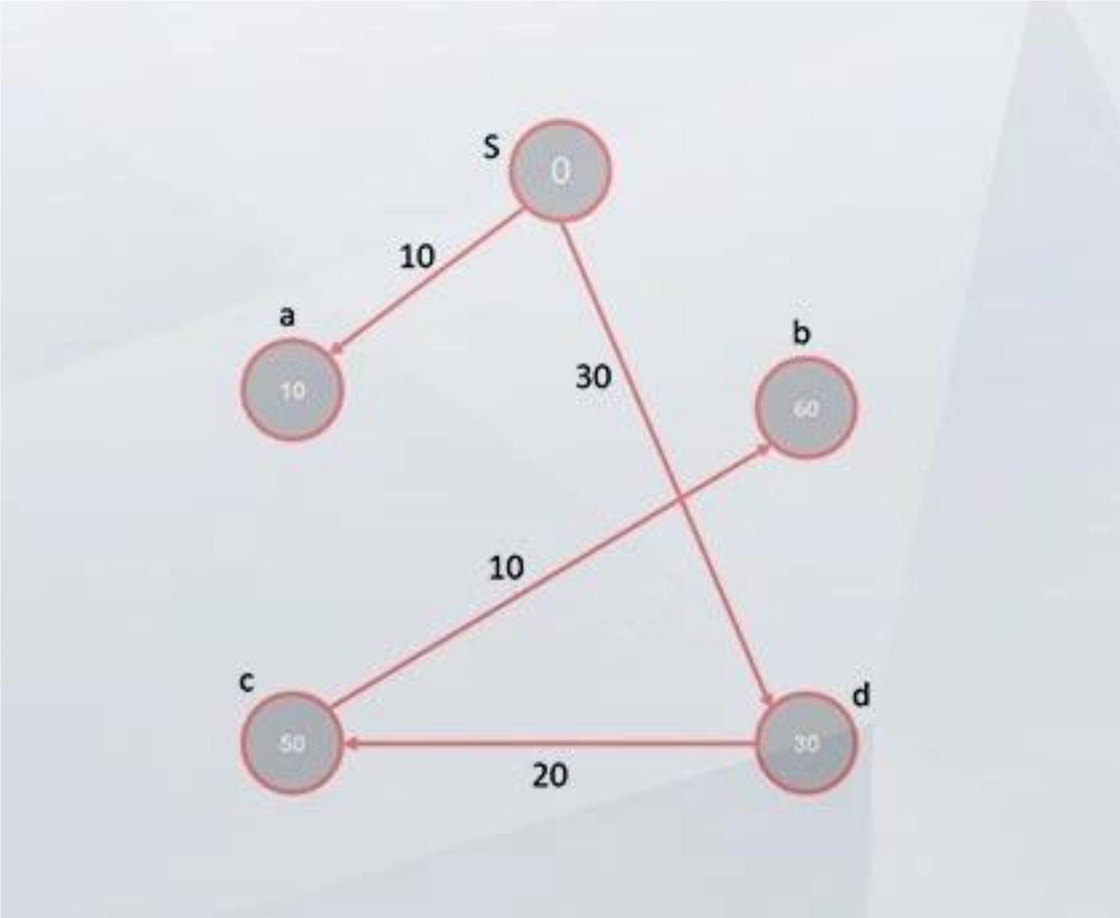
اکنون با حذف گره c از لیست فاصله‌ها، الگوریتم دایجسترا خاتمه میابد و در نهایت گراف و لیست زیر را خواهیم داشت.

	S	a	b	c	d
مرحله 1	0	∞	∞	∞	∞
مرحله 2		10	100	∞	30
مرحله 3			100	60	30
مرحله 4			90	50	
			60		

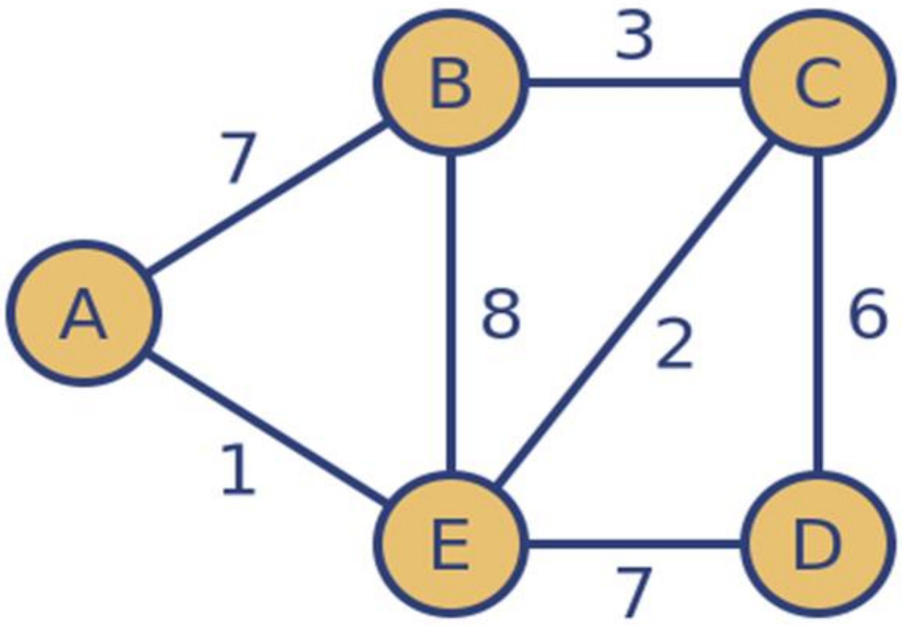
فاصله تا منبع

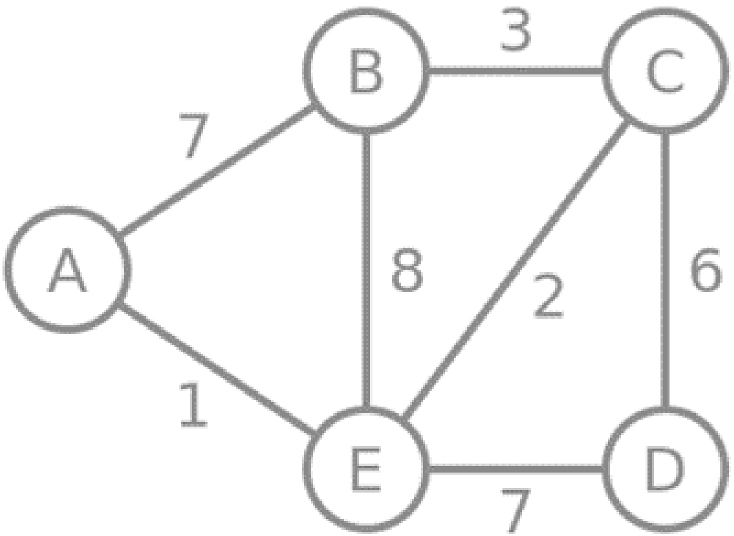


در گراف بدست آمده یال‌های قرمز رنگ کوتاه‌ترین مسیرها از گره منبع تا هر یک از گره‌های گراف هستند.

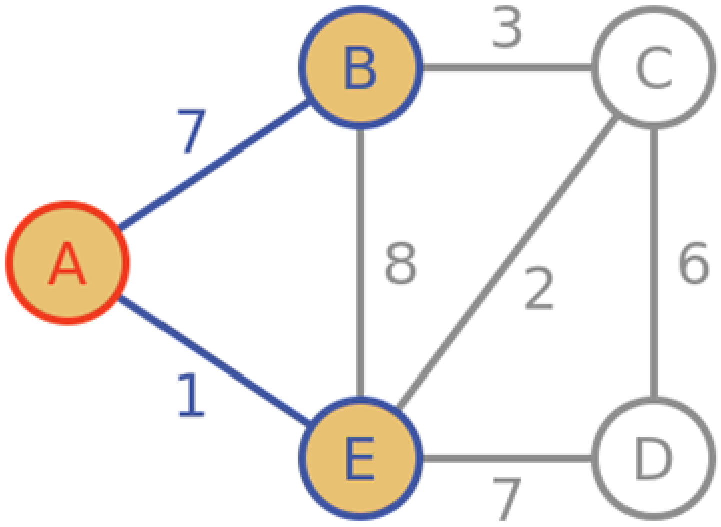


مثال دوم : محاسبه گراف و مسیر کوتاه‌ترین مسیر این گراف، از راس A

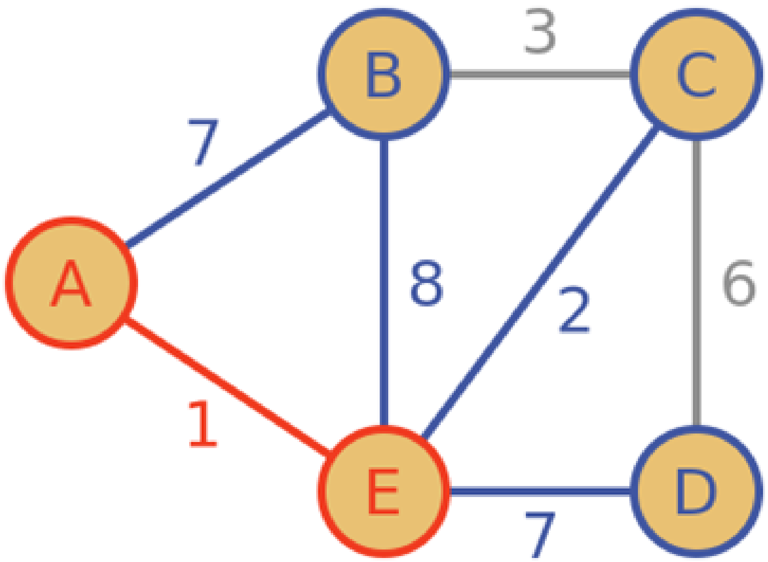




	Vis	Dist	Prev
A	0	∞	-
B	0	∞	-
C	0	∞	-
D	0	∞	-
E	0	∞	-

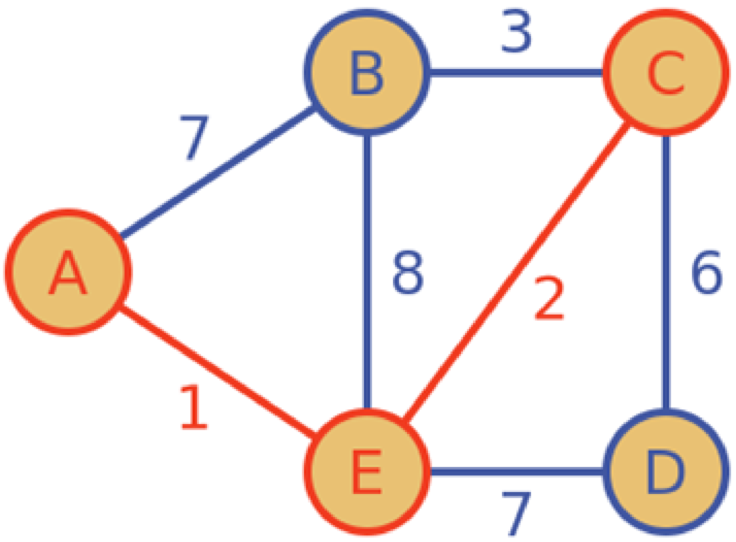


	Vis	Dist	Prev
A	1	0	-
B	0	7	A
C	0	∞	-
D	0	∞	-
E	0	1	A



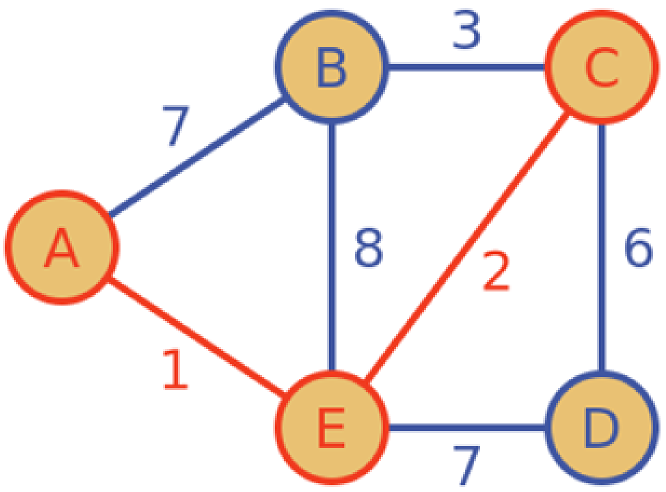
	Vis	Dist	Prev
A	1	0	-
B	0	7	A
C	0	3	E
D	0	8	E
E	1	1	A

در مرحله بعد، بار دیگر همه راس‌ها بازدیدنشده را بررسی می‌کنیم و آن راسی که کوچک‌ترین فاصله موقت را دارد به عنوان نقطه شروع انتخاب می‌کنیم. از جدول مرحله قبل، راس‌ها B، C و D بازدید نشده‌اند که در بین آن‌ها C کوچک‌ترین فاصله (۳) را دارد. پس C را به عنوان راس بازدیدشده علامت می‌زنیم.



	Vis	Dist	Prev
A	1	0	-
B	0	6	C
C	1	3	E
D	0	8	E
E	1	1	A

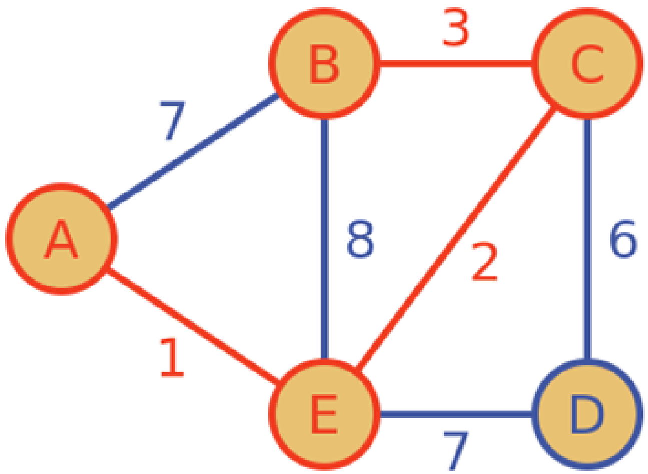
حال فاصله موقت همسایه‌های بازدیدنشده C را به‌روز می‌کنیم که B و D هستند. فاصله موقتی B از طریق C برابر با ۳ + ۳ = ۶ است. این فاصله از حاصل جمع فاصله اصلی C و وزن یال BC که هر دو برابر با ۳ هستند، بدست می‌آید. چون این فاصله جدید کمتر از فاصله قبلی (۷) است، فاصل موقت B را به ۶ و راس قبلی‌اش را هم به C تنظیم می‌کنیم. فاصله موقتی D از طریق C برابر با ۳ + ۶ = ۹ است. چون این فاصله بیشتر از فاصله فعلی است، آن را تغییر نمی‌دهیم:



	Vis	Dist	Prev
A	1	0	-
B	0	6	C
C	1	3	E
D	0	8	E
E	1	1	A

بازدید راس B

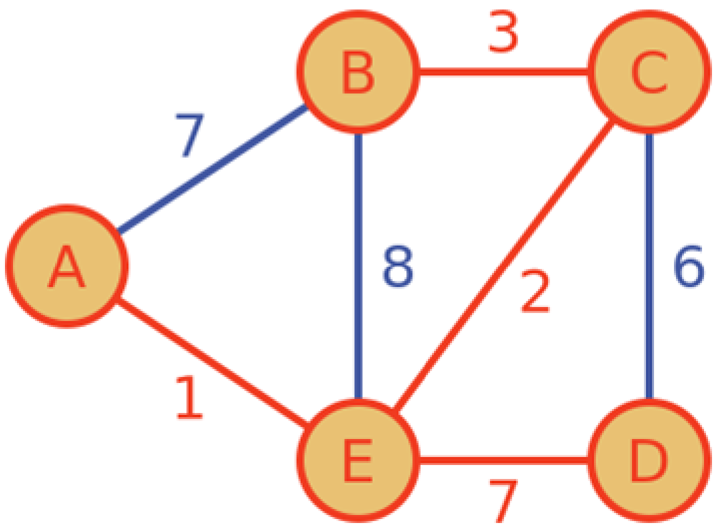
کار ما تقریبا تمام شده است. در واقع اکنون تنها راس‌ها B و D بازدید نشده‌اند و B کوچک‌ترین فاصله موقتی را دارد. بنابراین آن را با فاصله واقعی ۶ به عنوان راس بازدید شده علامت می‌زنیم. B هیچ همسایه بازدیدنشده‌ای ندارد بنابراین در این مرحله کار دیگری نباید انجام دهیم:



	Vis	Dist	Prev
A	1	0	-
B	1	6	C
C	1	3	E
D	0	8	E
E	1	1	A

بازدید راس D

اکنون تنها یک راس بازدیدنشده باقی مانده که D است. بنابراین آن را با فاصله واقعی ۸ (چون راه بهتری برای رسیدن به آن نیست) به عنوان بازدید شده علامت می‌زنیم. چون هیچ همسایه بازدید نشده‌ای از D نیز باقی نمانده، کار ما تمام است:



	Vis	Dist	Prev
A	1	0	-
B	1	6	C
C	1	3	E
D	1	8	E
E	1	1	A

درخت کوتاه‌ترین مسیر

نتیجه اجرای الگوریتم دایجسترا یک درخت است که می‌تواند برای محاسبه مسیر و فاصله هر راس از راس مبدا، یعنی A استفاده شود. بنابراین، به عنوان مثال، کوتاه‌ترین مسیر رسیدن به D برابر با AED با فاصله ۸ است و کوتاه‌ترین مسیر رسیدن به B برابر با AECB با طول ۶ است: