به نام خدا

مقدمه مؤلف

در این کتاب سؤالات مراجعی همچون CLRS ، MIT و Manber و ... پاسخ داده شده‌اند. سعی شده است سؤالات تکراری نباشند و بنابراین تست‌های کنکورها در این کتاب گردآوری نشده‌اند. توصیه می‌شود ابتدا کتاب طراحی الگوریتم چاپ ۱۰ به بعد و ساختمان داده چاپ ۱۱ به بعد پوران پژوهش مطالعه شوند و تست‌های آن دو کتاب پاسخ داده شوندو سپس سؤالات این کتاب مطالعه شود. این کتاب حاصل ۱۶ سال تدریس و مطالعه است و سعی شده است کتاب بدون خطا باشد. ولی وجود خطاهای سهوی طبیعی است. از شما خوانندهٔ گرامی تقاضا دارم هرگونه انتقاد و پیشنهاد راجع به این کتاب و سایر کتاب‌های اینجانب دارید را با بنده در میان بگذارید. لازم به ذکر است کتاب‌های اینجانب خطای علمی ندارند و پاسخ سؤالات دقیق و بدون اشکال علمی هستند. اگر مطلبی را متوجه نمی‌شوید، دلیل بر نادرستی آن مطلب نیست بلکه احتمالاً نیاز است که بیشتر مطالعه کنید و هنوز در آن مبحث علم کافی ندارید.

هادی یوسفی – بهار ۱۳۹۸
hadi_yusefi@yahoo.com
۰۹۱۲۱۷۸۸۵۳۴

فهرست مطالب

فصل اول. تحلیل الگوریتم‌های غیربازگشتی – نمادهای مجانبی – آنالیز استهلاکی ۱
فصل دوم. بازگشتی و تقسیم و غلبه ۲۱
فصل سوم. درهم‌سازی و جستجو ۵۳
فصل چهارم. مرتب‌سازی و مرتبه‌های آماری ۶۹
فصل پنجم. آرایه، صف، پشته و لیست پیوندی ۱۰۳
فصل ششم. درخت ۱۱۷
فصل هفتم. هیپ ۱۲۹
فصل هشتم. درخت‌های جستجو ۱۳۷
فصل نهم. حریصانه و پویا ۱۵۷
فصل دهم. گراف ۱۸۷
فصل یازدهم. سئوالات تکمیلی ۲۳۵

فصل ۱۰ گراف

بخش اول. سؤال‌ها

۱- چند تا از عبارات زیر صحیح است؟

a) گراف وزن‌دار $G=(V,E)$ با وزنهای متمایز و $S \subseteq V$ مفروض است. فرض کنید یال (u,v) یال با وزن مینیمم بین هر راس S و هر راس $V-S$ باشد. آنگاه MST (درخت پوشای مینیمم) گراف G باید شامل (u,v) باشد.

b) فرض کنید T یک MST از G است. آنگاه برای هر جفت راس s و t،کوتاهترین مسیر از s به t در G، همان مسیر از s به t در T است.

c) ممکن است DFS روی گراف جهت‌دار که حداقل یک یال دارد، اصلاً یال درختی تولید نکند.

۱) ۰ ۲) ۱ ۳) ۲ ۴) ۳

۲- یک مجموعه مستقل از گراف $G=(V,E)$، یک زیر مجموعه $V' \subseteq V$ از رئوس است که برای هر دو راس $u,v \in V'$ داشته باشیم $(u,v) \notin E$ یعنی بین رئوس مجموعه مستقل، یالی وجود ندارد. یک مجموعه مستقل ماکسیمال، مجموعه مستقلی مثل V' است طوری که برای هر $v \in V-V'$ مجموعه $V' \cup \{v\}$ مستقل نباشد یعنی هر راس که در V' نیست با راسی در V' مجاور باشد. با چه مرتبه‌ای می‌توان یک مجموعه مستقل ماکسیمال برای گراف $G=(V,E)$ یافت؟

۱) $O(|V|+|E|)$

۲) $O(|V|)$

۳) $O(|V|.|E|)$

۴) این مسئله NP – Complete است وهنوز راه حل چند جمله‌ای برای آن یافت نشده است.

۳- **یک $k-\text{coloring}$ از گراف غیر جهت دار $G=(V,E)$ تابع $f:V\rightarrow\{۱,۲,...,k\}$ است که برای هر یال $E\in(u,v)$، $f(u)\neq f(v)$. به عبارت دیگر به هر راس یـک رنـگ (یـک عدد از $\{۱,۲,...,k\}$) نسبت داده شود طوری که رئوس مجاور، رنگ متفاوت داشته باشند. با چه مرتبه‌ای می‌توان یک $(d+۱)-\text{coloring}$ برای گراف G که ماکزیمم درجه رئوسش d است، یافت ؟**

۱) $O(|V|+|E|)$

۲) $O(|V|)$

۳) $O(|V|.|E|)$

۴) این مسئله $NP-\text{Complete}$ است و هنوز راه حل چند جمله‌ای بـرای آن یافـت نـشده است.

۴- **گراف جهت‌دار G که هر یال $(u,v)\in E$ دارای وزن $w(u,v)$ (مثبت یا منفی) و همچنین رنـــگ $\text{color}(u,v)\in\{\text{red},\text{blue}\}$ اســـت را در نظـــر بگیریـــد. هزینـــه مـــسیر $P=v_۱\rightarrow v_۲\rightarrow....\rightarrow v_k$ برابر است با مجموع وزن یالهای مسیر بعـلاوه ۵ بـرای هـر جفت یال مجاوری که همرنگ نیستند. یعنی در طی مسیر اگر از یک یال بـه یـال بارنـگ متفاوت برویم، ۵ واحد هزینه اضافی داریم. یافتن کم هزینه‌ترین مسیر بین دو راس s و t از چه مرتبه‌ای است؟(n: تعداد رأس. e تعداد یال)**

۱) $O(n+e)$ ۲) $O(e.\lg n)$ ۳) $O(n.e)$ ۴) نمایی است.

۵- **در یک DAG می‌خواهیم بررسی کنیم تعداد مسیرهای از s به t زوج است یا فرد. بـا چـه مرتبه‌ای می‌توان این عمل را انجام داد؟**

۱) $O(n.e)$ ۲) $O(n+e)$ ۳) $O(e)$ ۴) $O(n)$

۶- **الگوریتم مقابل را در نظر بگیرید: «گراف غیر جهت‌دار همبند وزن‌دار G باوزن‌های متمـایز داده شده است. دائماً سنگین‌ترین یال را حذف کن به شرطی که گراف ناهمبند نـشود، تـا زمانیکه چنین یالی وجود نداشته باشد» آنگاه: (e تعداد یال، n تعداد رأس)**

۱) این الگوریتم حتماً MST (درخت پوشای مینیمم) برای G می‌یابـد و بـا مرتبـه $O(e^۲)$ قابل پیاده‌سازی است.

۲) این الگوریتم لزوماً MST نمی‌یابد و خروجی ممکن است سیکل داشته باشد.

۳) این الگوریتم MST نمی‌یابد و خروجی ممکن است ناهمبند است.

۴) این الگوریتم لزوماً MST می‌یابد ولی زمان آن نمایی است.

۷- **در گراف $G=(V,E)$، حداقل تعداد یال بین دو راس u و v را فاصله بین u و v گویند و با $dist(u,v)$ نشان می‌دهیم. قطر گراف G ماکزیمم فاصله بین هر جفت راس گراف است و با $diam(G)$ نشان می‌دهیم. $apd(G)$ میانگین فاصله‌ها بین هر جفت راس است یعنی:**

$$apd(G)=\frac{\sum_{\{u,v\}\subseteq V} dist(u,v)}{\binom{n}{۲}}$$

آنگاه:

a) **وجود دارد عدد طبیعی c که برای هر گراف همبند G داریم** $\frac{diam(G)}{apd(G)}\le c$

b) **برای هر گراف G داریم** $apd(G)=diam(G)$.

۱) فقط a درست است. ۲) فقط b درست است.

۳) b,a درست هستند. ۴) b,a نادرست هستند.

۸- **فرض کنید G یک گراف جهت‌دار است. گراف G' را به این صورت می‌سازیم: هر رأس $u'\in G'$ نمایانگر یک مولفه متصل قوی (SCC) از G است. یال (u',v') در G' وجود دارد اگر یک یال در G از SCC مربوط به u' به SCC مربوط به v' وجود داشته باشد آنگاه G':**

۱) ممکن است سیکل داشته باشد.

۲) حتماً سیکل دارد.

۳) حتماً سیکل ندارد (dag است).

۴) نمی‌توان اظهار نظر کرد.

۹- **کدام عبارات صحیح هستند؟**

a) **فرض کنید G گراف فاقد سیکل منفی است و از رأس s به رأس t مسیر وجود دارد. اگر کوتاه‌ترین مسیر از s به t شامل k تا یال باشد، آنگاه بعد از k امین مرحله اجرای بلمن‌فورد، کوتاه‌ترین مسیر s به t را می‌یابد.**

b) **اگر بلمن‌فورد بعد از k امین مرحله کوتاه‌ترین مسیر s به t را بیابید آنگاه کوتاه‌ترین مسیر از s به t شامل حداکثر k تا یال است.**

c) **عمق یک درخت BFS از یک گراف غیر جهت‌دار G با شروع از یک رأس دلخواه v، قطر گراف است.**

۱) a ۲) a,b ۳) b,c ۴) a,c

۱۰- کدام گزینه صحیح است؟

۱) الگوریتم دایجسترا ممکن است برای گراف با یال منفی، هیچ‌گاه خاتمه نیابد.

۲) فرض کنید $G=(V,E,W)$ یک گراف وزن‌دار جهت‌دار است و X کوتاه‌ترین مسیر s به t است که $s,t \in V$. اگر ما وزن همه یال‌ها را دو برابر کنیم یعنی $w'(e)=۲w(e)$ برای هر $e \in E$، آنگاه X همچنان کوتاه‌ترین مسیر s به t در گراف $G=(V,E,W')$ است.

۳) اگر گراف جهت‌دار G سیکل داشته باشد ولی با حذف یک یال بدون سیکل شود، آنگاه DFS دقیقاً یک یال back خواهد داشت.

۴) گزینه ۲ و ۳.

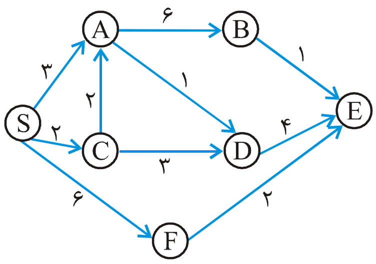
۱۱- ترتیب ملاقات رئوس توسط دایجسترا در گراف زیر کدام است؟ (چپ به راست)

۱) SCADEFB

۲) SCADFEB

۳) SCAFDEB

۴) SCAFEDB



۱۲- با چه مرتبه‌ای می‌توان در یک گراف جهت‌دار بدون سیکل (DAG) وزن‌دار، طولانی‌ترین مسیرهای ساده را نسبت به یک راس شروع یافت؟ (گراف n راس و e یال دارد)

۱) $\theta(e)$ ۲) $\theta(n)$

۳) $\theta(n+e)$ ۴) این مسئله راه حل چند جمله‌ای ندارد.

۱۳- یک یال e در یک گراف غیر جهت‌دار $G(V,E)$ را یک یال decycling گویند اگر حذف آن، گراف را بدون سیکل کند. کدام عبارات صحیح هستند؟ (فرض می‌کنیم گراف سیکل دارد)

a) شرط لازم و کافی برای وجود یال decycling آن است که گراف، یک جنگل بعلاوه یک یال اضافه باشد.

b) شرط لازم و کافی برای وجود یال decycling آن است که DFS فقط یک یال back تولید کند.

c) می‌توان با کمک DFS و با مرتبه $O(|V|)$ همه یالهای decycling را یافت.

۱) b,c ۲) a,c ۳) a,b ۴) a,b,c

۱۴- کدام گزینه نادرست است؟

۱) فرض کنید وزن همه یالهای گراف غیرجهت‌دار G مثبت و متمایز است. T درخت پوشای مینیمم G است. حال اگر وزن همه یالهای گراف را مربع کنیم، T همچنان پوشای مینیمم G است.

۲) فرض کنید وزن همه یالهای گراف غیرجهت‌دار G مثبت و متمایز است. p کوتاهترین مسیر از رأس s به t است. حال اگر وزن همه یالهای گراف را مربع کنیم، p همچنان کوتاهترین مسیر از رأس s به t است.

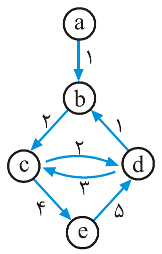
۳) اگر گراف دارای یالهای با وزن منفی باشد ولی سیکل منفی نداشته باشد آنگاه الگوریتم‌های کراسکال و پریم جواب درست می‌دهند.

۴) اگر گراف دارای سیکل منفی باشد آنگاه الگوریتم‌های کراسکال و پریم جواب درست می‌دهند.

۱۵- فرض کنید G گراف جهت‌دار وزن داراست. می‌خواهیم رأس مرکزی گراف را بیابیم. منظور از رأس مرکزی، رأسی است که میزان خروج از مرکزش مینیمم باشد. میزان خروج از مرکز رأس v عبارت است از:

$\max_{w \in V}\{\text{min Length of a path frow w to v}\}$

مثلاً برای گراف مقابل میزان خروج از مرکز هر رأس نوشته شده است: b۶ ،c۸ ،d۵ ،e۷ **و** a∞ **پس مرکز این گراف d است. کدام گزینه برای یافتن مرکز گراف مناسب است.**



۱) فلوید اجرا کنید، در هر ستون ماکزیمم را بیابید، رأسی که مینیمم عدد را دارد جواب است.

۲) فلوید اجرا کنید، در هر سطر ماکزیمم را بیابید، رأسی که مینیمم عدد را دارد جواب است.

۳) فلوید اجرا کنید، در هر سطر مینیمم را بیابید، رأسی که مینیمم عدد را دارد جواب است.

۴) فلوید اجرا کنید، در هر ستون مینیمم را بیابید، رأسی که ماکزیمم عدد را دارد جواب است.

۱۶- با توجه به تست قبل، مرتبه زمانی یافتن مرکز گراف n رأسی و e یالی کدام است؟

۱) $O(n^2)$ ۲) $O(n^4)$ ۳) $O(n^3)$ ۴) $O(n^3 Lgn)$

۱۷- **فرض کنید لیست مجاورت یک گراف $G=(V,E)$ را داشته باشیم. می‌خواهیم الگوریتمی تهیه کنیم که ترانهاده این گراف را بدست آورد. در این صورت این کار را در کدام مرتبه زمانی می‌توان انجام داد؟**

۱) $O(|V|)$ ۲) $O(|E|)$ ۳) $O(|V|+|E|)$ ۴) $O(|V|.|E|)$

۱۸- **در یک گراف همبند، یال bridge ، یالی است که با حذف آن گراف، ناهمبند می‌شود. می‌خواهیم در گراف همبند $G=(V,E)$ تعداد یالهای bridge را بیابیم. در این صورت این کار را در چه مرتبه زمانی می‌توان انجام داد؟**

۱) $O(|V|)$ ۲) $O(|E|)$ ۳) $O(|V|^۲)$ ۴) $O(|V|.|E|)$

۱۹- **گر اف G با ماتریس مجاورتی داده شده است. (۱) BFS را اجرا کرده‌ایم آنگاه گزینه صحیح کدام است؟**

$$A=\begin{array}{c|c} & \begin{matrix} ۱ & ۲ & ۳ & ۴ & ۵ & ۶ \end{matrix} \\ \begin{matrix} ۱ \\ ۲ \\ ۳ \\ ۴ \\ ۵ \\ ۶ \end{matrix} & \begin{bmatrix} ۰ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ \end{bmatrix} \end{array}$$

۱) $\#\text{forward edge} = ۲$, $\#\text{back edge} = ۳$

۲) $\#\text{bac kedge} = ۲$, $\#\text{cross edge} = ۳$

۳) $\#\text{cross edge} = ۲$, $\#\text{back edge} = ۱$, $\#\text{forword edge} = ۲$

۴) $\#\text{cross edge} = ۲$, $\#\text{back edge} = ۳$

۲۰- **فرض کنید T درخت DFS حاصل از گراف غیرجهت‌دار همبند G است برای هر راس v در T، Pre (v) تعداد نودهای ملاقات شده تا نود v (با خود v) در طی یک پیمایش پیش ترتیب T باشد و Post (v) تعداد نودهای ملاقات شده تا نود v (با خود v) در طی یک پیمایش پس ترتیب T باشد.**
کوچکترین جد مشترک v,u در T راسی است مثل w در T به طوری که w جد v,u باشد و هیچ فرزند w جد v,u نباشد.
فرض کنید (u,v) یک یال G باشد که در T نیست به طوری که Pre(u) < Pre (v). کدام یک از جملات زیر راجع به v,u صحیح است.

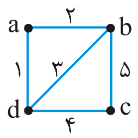
I. post (u) < post (v)
II. u is an ancestor of v in T
III. if w is the lowest common ancestor of u and v in T then w=u

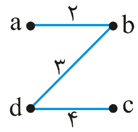
۱) I ۲) I , III ۳) II , IIII ۴) هر سه

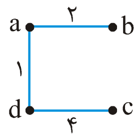
۲۱- گراف G با لیست مجاورتی نمایش داده شده است، می‌خواهیم تشخیص دهیم این گراف دوبخشی (bipartite) است یا خیر، مرتبه الگوریتم ارائه‌شده کدام است؟ (n تعداد راس و e تعداد یال است)

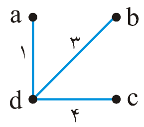
۱) $O(e+n)$ ۲) $O(n^۲)$ ۳) $O(en)$ ۴) $O(e\lg n)$

۲۲- کدام درخت، درخت پوشای گلوگاه گراف مقابل است؟



۱) 

۲) 

۳) 

۴) هر سه مورد صحیح است.

۲۳- فرض کنید $G=(V,E)$ یک گراف همبند است و $u \in V$ ، درخت DFS را با شروع از u بدست آورده ایم و سپس درخت BFS را با شروع از u بدست آورده ایم. در چه صورتی این دو درخت یکی هستند؟

۱) G خود درخت باشد ۲) G کامل باشد

۳) G همیلتنی باشد ۴) هر ۳ گزینه امکان پذیر است.

۲۴- برای یافتن درخت پوشای مینیمم (MST) ، با روش تقسیم و غلبه (بازگشتی) رئوس گراف را دو قسمت می کنیم سپس در هر قسمت به صورت بازگشتی MST را می یابیم و سپس سبک‌ترین یالی که دو قسمت را به هم وصل می کند، انتخاب می‌کنیم، مرتبه این الگوریتم کدام است؟ (n تعداد راس و e تعداد یال)

۱) $O(n \log n)$ ۲) $O(e \log n)$

۳) $O(n^۲)$ ۴) چنین الگوریتمی نمی‌تواند MST بیابید.

۲۵- در گراف G که DAG است می‌خواهیم وزن سنگین‌ترین مسیر از راس داده شده s به هر یک از رئوس دیگر گراف را بیابیم. در واقع می‌خواهیم خروجی آرایه $d[۱..n]$ باشد که در $d[i]$ وزن سنگین‌ترین مسیر از راس s به راس i ذخیره شود. سریع‌ترین الگوریتم از چه زمانی است؟

۱) $\theta(n+e)$ ۲) $\theta(n^۲)$

۳) $\theta(e+n^۲)$ ۴) الگوریتم چند جمله‌ای ندارد.

۲۶- در جستوی BFS یک گراف چند تا از گزاره‌های زیر درست است؟

- حافظه آن روی یک DAG با n راس و حداکثر طول مسیر L از مرتبه $O(L)$ است.
- زمان اجرای آن برای DAGها به جای $O(n+e)$، $O(n)$ است.
- مقدار $d[u]$ که در خروجی به راس u داده می‌شود مستقل از ترتیب رئوس در لیست مجاورت است.

۱) ۰ ۲) ۱ ۳) ۲ ۴) ۳

۲۷- گراف وزن‌دار G با n راس و e یال داده شده است. یک یال خاص $q=(u,v)$ نیز از آن علامت خورده است. می‌خواهیم کم‌وزن‌ترین درخت فراگیری را بیابیم که شامل یال q باشد. اگر بهترین الگوریتم برای یافتن MST گراف، الگوریتم x از زمان $O(T)$ باشد، کدام گزینه درباره بهترین زمان اجرای یک الگوریتم برای کار خواسته شده درست است؛ فرض کنید ما از ساختار الگوریتم x خبر نداریم و همیشه باید به این الگوریتم یک گراف دقیق و مشخص (با لیست مجاورت) بدهیم.

۱) $O(n+T)$ ۲) $O(n+e+T)$ ۳) $O(n.T)$ ۴) $O((n+e).T)$

۲۸- گراف ساده، وزن‌دار و بدون جهت G با n راس و e یال (با وزن‌های متفاوت و مثبت) داده شده است. برای یک درخت فراگیر T از G که یال‌های آن را با $E(T)$ نشان می‌دهیم، مقدار «زشتی» یا $U(T)$ را طبق فرمول زیر محاسبه می‌کنیم: $U(T)=\sum_{e\in E(T)} w(e)-\max_{e\in E(T)} w(x)$ که $w(e)$، وزن یال e است. به عبارت ساده، زشتی یک درخت فراگیر برابر جمع وزن تمامی یال‌های آن، به جز سنگین‌ترین یال‌اش است. می‌خواهیم زیباترین درخت فراگیر (درختی با کمترین مقدار زشتی) را در G بیابیم. اگر بهترین الگوریتم برای یافتن MST در این گراف، الگوریتم ناشناخته x با زمان $\theta(T)$ باشد، گزینه مناسب درباره بهترین زمان اجرای یک الگوریتم برای کار خواسته شده کدام است؟

۱) $\theta(T)$ ۲) $\theta(n+e+T)$ ۳) $\theta(n.T)$ ۴) $\theta((n+e).T)$

۲۹- فرض کنید G یک گراف ساده، بدون جهت و وزن‌دار با وزن‌های مثبت است و k تعداد MSTهای مختلف G است. دو MST مختلف‌اند اگر دست کم در یک یال اختلاف داشته باشند. چند تا از سه گزاره زیر صحیح است؟

a) اگر وزن یال‌های G دوبدو متمایز باشند آنگاه $k=۱$

b) اگر $k=۱$ ناگزیر وزن یال‌های G دوبدو متفاوت است.

c) اگر دست کم دو تا از یال‌های G وزن یکسان داشته باشند $k\geq ۲$

۱) ۰ ۲) ۱ ۳) ۲ ۴) ۳

۳۰- فرض کنید یال $e=(x,y)$ بیشترین وزن را در میان یال‌های سازنده دور C از گراف ساده و هم بند $G=(V,E)$ دارد. گراف $G'=(V,E-\{e\})$ را در نظر بگیرید. فرض کنید که S_G مجموعه تمام درخت‌های فراگیر کمینه G باشد و $S_{G'}$ مجموعه تمام درخت‌های فراگیر کمینه G' باشد. کدام گزینه همواره درست است؟

۱) $S_G \cap S_{G'} = S_G$

۲) $S_G \cap S_{G'} = S_{G'}$

۳) $S_G \cap S_{G'} = \phi$

۴) هیچکدام

۳۱- فرض کنید G یک گراف غیرجهت‌دار حاوی n راس است و دو راس s و t وجود دارد که طول کوتاه‌ترین مسیر بین s و t بیشتر از $\frac{n}{2}$ است. آنگاه:

۱) حتماً رأسی مثل v وجود دارد که G-v فاقد مسیر از s به t است، وبا مرتبه $O(n+e)$ می‌توان این راس را یافت.

۲) حتماً رأسی مثل v وجود دارد که G-v فاقد مسیر از s به t است، و با مرتبه $O(n.e)$ می‌توان این راس را یافت.

۳) ممکن است راسی مثل v وجود داشته باشد که G-v فاقد مسیر از s به t است و با مرتبه $O(n+e)$ می‌توان متوجه شد.

۴) ممکن است راسی مثل v وجود داشته باشد که G-v فاقد مسیر از s به t است و با مرتبه $O(n.e)$ می‌توان متوجه شد.

۳۲- با چه مرتبه‌ای می‌توان، تعداد کوتاهترین مسیرها را از رأس v به همه رئوس محاسبه کرد؟

۱) $O\big(n(n+e)\big)$ ۲) $O(n+e)$

۳) $O(n.e)$ ۴) $O(e^{۲})$

۳۳- گراف غیروزن‌دار $G=(V,E)$ که برخی یال‌های آن جهت‌دار و برخی غیرجهت‌دار هستند را در نظر بگیرید. فرض کنید یال‌های جهت‌دار در گراف G با هم سیکل تشکیل نمی‌دهند. با چه مرتبه‌ای می‌توان به یال‌های غیرجهت‌دار، جهت نسبت داد طوری که گراف G کاملاً بدون سیکل باشد؟ (بهترین گزینه را انتخاب کنید)

۱) $O(n.e)$ ۲) $O(n)$

۳) $O(e)$ ۴) $O(n+e)$

۳۴- گراف غیرجهت‌دار وزن‌دار G با وزن‌های مثبت را در نظر بگیرید. فرض کنید یال‌ها دو نوع هستند: ناهموار و هموار. در واقع به هر یال e مقدار بولین $r[e]$ نسبت داده شده است که نشان‌دهنده هموار یا ناهموار بودن یال است. می‌خواهیم کوتاهترین مسیر از رأس s به رأس t را بیابیم به شرطی که در این مسیر حداکثر یک یال ناهموار وجود داشته باشد. کدام گزینه صحیح است؟

۱) با زمان خطی تغییری در گراف می‌دهیم و سپس با الگوریتم دایجسترا می‌توان مسئله را حل کرد.

۲) با زمان خطی تغییری در گراف می‌دهیم و سپس با الگوریتم BFS می‌توان مسئله را حل کرد.

۳) این مسئله راه حل چندجمله‌ای ندارد.

۴) با دو بار اجرای الگوریتم دایجسترا می‌توان مسئله را حل کرد.

۳۵- لیست مجاورتی گراف جهت‌دار $G=(V,E)$ داده شده است. با چه مرتبه‌ای می‌توان درجه خروجی هر راس را یافت؟ با چه مرتبه‌ای می‌توان درجه ورودی هر رأس را یافت؟ ($|V|=n$ و $|E|=e$)

۱) هر دو $\theta(n+e)$ ۲) هر دو $\theta(n)$

۳) هر دو $\theta(e)$ ۴) درجه خروجی $\theta(n+e)$ و درجه ورودی $\theta(ne)$

۳۶- مربع گراف جهت‌دار $G=(V,E)$ عبارت است از $G^2=(V,E^2)$ که $(u,v)\in E^2$ اگر و فقط اگر G شامل مسیری با حداکثر دو یال از u به v باشد. با چه مرتبه‌ای می‌توان از روی گراف G، گراف G^2 را بدست آورد؟ (بهترین گزینه را انتخاب کنید)

۱) با لیست مجاورتی $\theta(n+e)$ با ماتریس مجاورتی $\theta(n^2)$

۲) با لیست مجاورتی $\theta(n.e)$ با ماتریس مجاورتی $\theta(n^3)$

۳) با لیست مجاورتی $\theta(n.e)$ با ماتریس مجاورتی $\theta(n^{\lg 7})$

۴) با لیست مجاوری $\theta(n+e)$ با ماتریس مجاورتی $\theta(n^3)$

۳۷- ترانهاده‌ی گراف جهت‌دار $G=(V,E)$، گراف $G^T=(V,E^T)$ است که $E^T=\{(v,u)\in V\times V:(u,v)\in E\}$. به عبارتی جهت همه یال‌های G را برعکس کنیم آنگاه G^T حاصل می‌شود. با چه هزینه زمانی می‌توان G^T را بدست آورد؟

۱) با ماتریس مجاورتی $\theta(n^2)$ و با لیست مجاورتی $\theta(n.e)$

۲) با ماتریس مجاورتی $\theta(n)$ و با لیست مجاورتی $\theta(n+e)$

۳) با ماتریس مجاورتی $\theta(n^2)$ و با لیست مجاورتی $\theta(n+e)$

۴) با ماتریس مجاورتی $\theta(n)$ و با لیست مجاورتی $\theta(n.e)$

۳۸- با چه مرتبه‌ای می‌توان متوجه شد که گراف جهت‌دار G دارای «چاه فراگیر» (universal sink) هست یا خیر؟ چاه فراگیر رأسی با درجه خروجی ۰ و درجه ورودی $n-1$ است. فرض کنید گراف G با ماتریس مجاورتی پیاده‌سازی شده است؟

۱) $O(\lg n)$ ۲) $O(n.\lg n)$ ۳) $O(n)$ ۴) $O(n^2)$

۳۹- ماتریس برخورد گراف جهت‌دار $G=(V,E)$ که طوقه (Self loop) ندارد، ماتریس $B=[b_{ij}]_{n\times e}$ است به طوری که:

$$b_{ij}=\begin{cases}-1 & \text{اگر یال } j \text{ از رأس } i \text{ خارج شود}\\ 1 & \text{اگر یال } j \text{ به رأس } i \text{ وارد شود}\\ 0 & \text{در غیر این صورت}\end{cases}$$

درایه‌های $B.B^T$ چه مفهومی را نشان می‌دهند؟ (B^T ترانهاده B است)

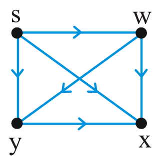
۱) جمع درایه‌های $B.B^T$ دو برابر تعداد یال‌های گراف است.

۲) درایه‌های قطر اصلی $B.B^T$ درجات رئوس گراف هستند و درایه $[i,j]$ که $i\neq j$، نشان می‌دهد چند تا یال بین رأس i و j هست.

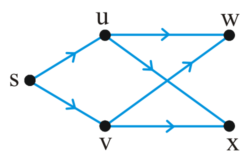
۳) جمع درایه‌های $B.B^T$ دو برابر جمع درجات ورودی و خروجی است.

۴) هیچ مفهوم خاصی ندارد.

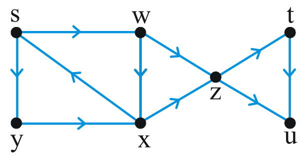
۴۰- در کدام گراف با شروع از مبدأ s، درخت کوتاهترین مسیری وجود دارد که با BFS نمی‌توان این درخت را بدست آورد؟



۱)



۲)



۳)

۴) BFS می‌تواند با مبدأ s هر درخت کوتاهترین مسیری را بیابد.

۴۱- قطر (diameter) درخت $T=(V,E)$ برابر $\max_{u,v\in V}\delta(u,v)$ است، یعنی بزرگترین مقدار در بین همه کوتاهترین مسیرها در درخت. بهترین گزینه برای یافتن قطر درخت T کدام است؟ (درخت با لیست مجاورتی پیاده‌سازی شده و بدون وزن است)

۱) با شروع از یک رأس دلخواه BFS اجرا کنیم، بزرگترین مقدار d(distance)، قطر درخت است.

۲) n بار و هر بار با شروع از یک رأس جدید BFS اجرا کنیم، بزرگترین مقدار d بدست آمده، قطر درخت است.

۳) با شروع از یک رأس دلخواه مثل s، BFS اجرا کنیم و آخرین رأسی که ملاقات می‌شود (دورترین رأس از s) را u می‌نامیم. سپس با شروع از u مجدداً BFS اجرا می‌کنیم و آخرین رأسی که ملاقات می‌شود v می‌نامیم. $d(u,v)$ (فاصله u تا v) برابر قطر گراف است.

۴) برای یافتن قطر نمی‌توان از BFS استفاده کرد و باید از بلمن فورد کمک بگیریم.

۴۲- کدام جملات صحیح هستند؟

a) **در گراف جهت‌دار G اگر مسیری از u به v وجود داشته باشد و $d[u]<d[v]$ آنگاه در جنگل حاصل از پیمایش عمقی گراف G، رأس v نواده رأس u خواهد بود.** ($d[u]$ **زمان ملاقات رأس u است** discovery time)

b) **در گراف جهت‌دار G اگر مسیری از u به v باشد آنگاه در هر پیمایش عمقی داریم** $d[v]\le f[u]$.

۱) فقط a ۲) فقط b ۳) a و b ۴) هیچکدام

۴۳- گراف جهت‌دار $G=(V,E)$ را متصل یکصرفه (Singly connected) گویند هرگاه به ازای هر دو راس u و v حداکثر یک مسیر ساده از u به v وجود داشته باشد. با چه روشی می‌توان تشخیص داد یک گراف جهت‌دار، متصل یکطرفه است؟ (مسیر ساده مسیری است که رأس و یال تکراری ندارد)

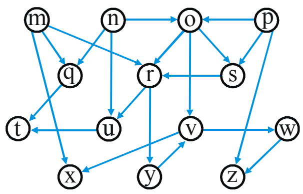
۱) با شروع از یک رأس دلخواه پیمایش عمقی اجرا کنیم. گراف متصل یکطرفه است اگر و فقط اگر پیمایش عمقی، یال cross و forward تولید نکند.

۲) با شروع از همه رئوس یعنی n بار پیمایش عمقی اجرا کنیم. گراف متصل یکطرفه است اگر و فقط اگر پیمایش عمقی، یال cross تولید نکند.

۳) با شروع از همه رئوس یعنی n بار پیمایش عمقی اجرا کنیم. گراف متصل یکطرفه است اگر و فقط اگر پیمایش عمقی، یال back تولید نکند.

۴) هیچکدام

۴۴- گراف جهت‌دار بدون سیکل G (dag) و دو رأس s و t داده شده‌اند. می‌خواهیم تعداد مسیرهای ساده از رأس s به رأس t را بشماریم. مثلاً در dag مقابل تعداد ۴ مسیر ساده از رأس P به رأس v وجود دارد: psryv و posryv و poryv و pov. با چه مرتبه‌ای می‌توان تعداد مسیرهای مورد نظر را یافت؟ (بهترین گزینه را انتخاب کنید)



۱) $O(n)$

۲) $O(e)$

۳) $O(n+e)$

۴) $O(n.e)$

۴۵- با چه مرتبه‌ای می‌توان تشخیص داد که گراف غیرجهت‌دار $G=(V,E)$ دارای سیکل است؟ (بهترین گزینه را انتخاب کنید.)

۱) $O(n)$ ۲) $O(e)$ ۳) $O(n+e)$ ۴) $O(n.e)$

۴۶- برای یافتن مولفه‌های متصل قوی (scc) گراف جهت‌دار G، روی گراف G پیمایش عمقی را انجام می‌دهیم و برای هر رأس زمان پایان (f) را ثبت می‌کنیم و سپس ترانهاده گراف G (G^T) را می‌یابیم و روی G^T مجدداً پیمایش عمقی ولی به ترتیب نزولی fها اجرا می‌کنیم؛ در جنگل حاصل از پیمایش، هر درخت، یک مولفه متصل قوی است. کدام عبارات صحیح هستند؟

a) می‌توان بجای اجرای پیمایش عمقی روی G^T به ترتیب نزولی fها، روی خود G پیمایش عمقی را به ترتیب صعودی fها انجام داد.

b) اگر یالی به گراف G اضافه شود آنگاه تعداد مولفه‌های متصل قوی ممکن است کاهش یابد.

۱) a ۲) b ۳) a,b ۴) هیچکدام

۴۷- تور اویلری در یک گراف جهت‌دار $G=(V,E)$ که متصل قوی (Strongly connected) است، سیکلی است که هر یال گراف را دقیقاً یک بار طی کند، هرچند ممکن است یک رأس بیش از یک بار ملاقات شود. کدام عبارات صحیح هستند؟

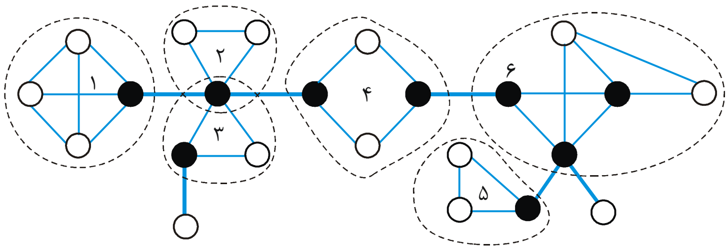
a) گراف G تور اویلری دارد اگر فقط اگر برای هر رأس $u \in V$ داشته باشیم indegree(u)=outdegree(u).

b) با مرتبه $O(e)$ می‌توان تور اویلری گراف G را در صورت وجود یافت.

c) چون گراف G متصل قوی است حتماً تور اویلری دارد.

۱) a و b ۲) a و c ۳) c و b و a ۴) هیچکدام

۴۸- فرض کنید $G=(V,E)$ یک گراف غیرجهت‌دار همبند است. یک رأس برشی یا نقطه انفصال (articulation point) رأسی است که حذف آن، گراف G را ناهمبند می‌کند. یک پل (bridge) یالی است که حذف آن (بدون حذف دو رأس آن) گراف را ناهمبند می‌کند. یک مولفه دوسو همبند (biconnected component) در گراف G، یک مجموعه ماکسیمال از یال‌هاست که هر دو یال در مجموعه در یک سیکل ساده مشترک قرار داشته باشند. شکل زیر این مفاهیم را نشان می‌دهد:



نودهای پررنگ، نقطه انفصال هستند. یال‌های پررنگ، یال برشی هستند. مولفه‌های دوسو همبند نیز مشخص شده‌اند و شماره خورده‌اند. با کمک پیمایش عمقی می‌توان، نقاط انفصال، پل‌ها و مولفه‌های دوسو همبند را یافت. اگر $G_\pi=(V,E_\pi)$ درخت حاصل از پیمایش عمقی گراف G باشد. چند عبارت صحیح هستند؟

a) ریشه G_π نقطه انفصال برای G است اگر و فقط اگر ریشه G_π حداقل ۲ فرزند داشته باشد.

b) فرض کنید v یک رأس غیر از ریشه G_π باشد. v یک نقطه انفصال در G است اگر و فقط اگر v دارای فرزندی مثل s باشد که از s یا نواده‌های s به اجداد محض v هیچ یال پشتی وجود نداشته باشد.

c) با توجه به تعریف مقابل، می‌توان $low[v]$ را برای هر رأس $v\in V$ با مرتبه $O(e)$ محاسبه کرد:

$$low[v]=\min\begin{cases} d[v] \\ d[w]:(u,w)\,\text{is a back edge for some descendant}\ u\ \text{of}\ v\end{cases}$$

d) همه نقاط انفصال را می‌توان با مرتبه $O(e)$ یافت.

e) یک یال پل است اگر و فقط اگر عضو هیچ سیکل ساده‌ای در G نباشد.

f) همه پل‌ها را با مرتبه $O(e)$ می‌توان یافت.

g) مولفه‌های دوسو همبند در G، یال‌های غیرپل G را افراز می‌کنند.

h) با الگوریتمی از مرتبه $O(e)$ می‌توان به هر یال e یک عدد مثبت به نام $bcc[e]$ نسبت داد طوری که $bcc[e]=bcc[e']$ اگر و فقط اگر e و e' عضو یک مولفه دوسو همبند باشند.

۱) ۵ ۲) ۶ ۳) ۷ ۴) ۸

۴۹- فرض کنید $G=(V,E)$ گراف همبند غیرجهت‌دار وزن‌دار است. برش (cut) $(S,V-S)$، یک افراز برای رئوس V است. گوییم یال (u,v) این برش را قطع می‌کند هرگاه یک رأس مثلاً u در S و رأس دیگر در V-S باشد. گوییم یک برش به مجموعه یـال‌هـای A احتـرام می‌گذارد، هرگاه هیچ یال A از برش عبور نکند. یک یال را یال سبک (light) بـرای بـرش گویند اگر وزن این یال در بین همه یال‌های عبور کننـده از بـرش، مینـیمم باشـد. کـدام عبارات صحیح هستند؟

a) فرض کنید $A \subseteq E$ و A زیرمجموعه یال‌هایی اسـت کـه حـداقل در یـک MST از G وجود دارند. فرض کنید برش $(S,V-S)$ به A احترام می‌گذارد و یـال (u,v) یـک یـال سبک عبورکننده از این برش است آنگاه یال (u,v) برای A آمن (safe) است. (امن یعنی یالی که به A اضافه شود و A همچنان خاصیت درخت مینیمم داشته باشد)

b) فرض کنید $A \subseteq E$ و A زیرمجموعه یال‌هایی است کـه حـداقل در یـک MST از G وجود دارند. فرض کنید برش $(S,V-S)$ به A احترام می‌گذارد و یال (u,v) یک یال امن عبورکننده از این برش برای A باشد. آنگاه (u,v) یک یال سبک برای این برش است.

c) اگر یال (u,v) در یک MST باشد آنگاه این یال یک یال سبک عبـور کننـده از یـک برش گراف است.

d) مجموعه زیر تشکیل یک MST می‌دهد:

{برش $(S,V-S)$ وجود دارد که یال (u,v) یک یال سبک عبـور کننـده از ایـن بـرش اسـت: $\{(u,v)$

e) گراف دارای MST منحصر به فرد است اگر و فقط اگر برای هر برش، یـک یـال سـبک منحصر به فرد از این برش عبور کند.

۱) a و b
۲) a و c
۳) b و c و d و e
۴) a و b و c و d و e

۵۰- مرتبه الگوریتم کراسکال در هر یک از حالات زیر کدام است؟

a) وزن یال‌ها در گراف اعداد صحیح در بازه از ۱ تا n (n تعداد رئوس) باشد.

b) وزن یال‌ها اعداد صحیح در بازه ۱ تا W است که W ثابت است.

۱) a) $O(e.\lg n)$ ، b) $O(e.\lg n)$
۲) a) $O(e.\alpha(n))$ ، b) $O(e.\alpha(n))$
۳) a) $O(e.\lg n)$ ، b) $O(e.\alpha(n))$
۴) a) $O(e.\alpha(n))$ ، b) $O(e.\lg n)$

۵۱- مرتبه الگوریتم پریم در هر یک از حالات زیر کدام است؟ (بهترین گزینه را انتخاب کنید)

a) **وزن یال‌ها اعداد صحیح ۱ تا n باشد.**

b) **وزن یال‌ها اعداد صحیح ۱ تا W باشد که W ثابت است.**

۱) a) $O(n^2)$ — b) $O(e.\lg n)$

۲) a) $O(e+n.\lg n)$ — b) $O(e.\lg n)$

۳) a) $O(e+n.\lg n)$ — b) $O(e)$

۴) a) $O(e)$ — b) $O(e)$

۵۲- فرض کنید وزن یال‌ها به طور یکنواخت در بازه $[0,1)$ توزیع شده است آنگاه:

۱) می‌توان کراسکال را از پریم سریعتر اجرا کرد.

۲) می‌توان پریم را از کراسکال سریعتر اجرا کرد.

۳) زمان اجرای هر دو برابر است.

۴) در شرایط مختلف ممکن است کراسکال یا پریم سریعتر باشد.

۵۳- فرض کنید MST گراف G را بدست آورده‌ایم. حال اگر یک رأس به G اضافه کنیم و از آن رأس تعدادی یال به رئوس G اضافه کنیم با چه مرتبه‌ای می‌توان MST گراف جدید را یافت؟ (بهترین گزینه را انتخاب کنید)

۱) $O(e.\lg n)$ ۲) $O(n^2)$ ۳) $O(n.\lg n)$ ۴) $O(n)$

۵۴- گراف غیرجهت‌دار، همبند و وزن‌دار $G=(V,E)$ با تابع وزن $\omega: E\to\mathbb{R}$ که وزن یال‌ها متمایز است و همچنین $|E|\ge|V|$ را در نظر بگیرید. درخت فراگیر کمینه‌ی دوم، درختی فراگیری است که جمع وزن‌های آن از MST بیشتر و از سایر درخت‌های پوشا، کمتر است. کدام عبارات صحیح هستند؟

a) **MST منحصر به فرد است ولی درخت فراگیر کمینه دوم، لزوماً منحصر به فرد نیست.**

b) **اگر T و T' به ترتیب اولین و دومین درخت فراگیر کمینه باشند آنگاه T و T' دقیقاً در یک یال تفاوت دارند یعنی یک یال مثل (u,v) از T حذف کنیم و یالی مثل $(x,y)\in E$ را به T اضافه کنیم آنگاه T' حاصل می‌شود.**

c) **با مرتبه $O(n+e)$ $(|V|=n, |E|=e)$ می‌توان درخت فراگیر کمینه دوم را یافت.**

۱) a و b ۲) c و a ۳) c و b ۴) a و b و c

۵۵- فرض کنید G یک گراف همبند غیرجهت‌دار و وزن‌دار با تابع وزن w است. سه الگوریتم زیر مجموعه یال‌های T را برمی‌گردانند. در کدام الگوریتم T یک MST برای گراف G است.

a.
```
Sort the edges into nonincreasing order of edge weights w
T = E
for each edge e, taken in nonincreasing order by weight
    if T − {e} is a connected graph
        T = T − {e}
return T
```

b.
```
T = φ
for each edge e, taken in arbitrary order
    if T ∪ {e} has no cycles
        T = T ∪ {e}
return T
```

c.
```
T = φ
for each edge e, taken in arbitrary order
    T = T ∪ {e}
    if T has a cycle c
        let e′ be a maximum-weight edge on c
        T = T − {e′}
return T
```

۱) a و c ۲) a و b ۳) b و c ۴) a و b و c

۵۶- کدام گزینه صحیح است؟

۱) تعداد یالِ گراف ساده جهت‌دار بدون دور با n رأس حداکثر $\binom{n}{2}$ است.

۲) حداکثر تعداد مؤلفه‌های هم‌بند قوی یک گراف جهت‌دار با n رأس برابر $\lceil \frac{n}{2} \rceil$ است.

۳) فرض کنید d تعداد مؤلفه‌های هم‌بند قوی گراف جهت‌دار G است. از گراف G، گراف G' را با همان رأس‌ها و با برداشتن جهت تمام یال‌ها می‌سازیم. اگر u تعداد مؤلفه‌های هم‌بندی گراف G' باشد آنگاه $u \geq d$ می‌باشد.

۴) فرض کنید در گراف جهت‌دار G با n رأس و e یال، به ازای هر ۲ رأس u و v، حداکثر یکی از یالهای $u \to v$ یا $v \to u$ وجود دارد. آنگاه تعداد دورهای متفاوتِ گراف G از مرتبه $O(n^2)$ است. دو دور c_1 و c_2 را متفاوت گویند هرگاه حداقل یک یال غیرمشترک داشته باشند.

۵۷- یک درخت پوشای گلوگاه T از یک گراف غیرجهت‌دارِ G، یک درخت پوشا از G است که وزن سنگین‌ترین یال آن در بین تمام درخت‌های پوشای G مینیمم باشد، یعنی سنگین‌ترین یالِ درخت پوشای گلوگاه T، از سنگین‌ترین یالِ سایر درخت‌های پوشای گراف G، سنگین‌تر نیست. گوییم مقدار درخت پوشای گلوگاه برابرِ با وزن سنگین‌ترین یال در T است. چند تا از عبارات زیر صحیح هستند؟

a) **یک درخت پوشای مینیمم، یک درخت پوشای گلوگاه است.**

b) **یک درخت پوشای گلوگاه، یک درخت پوشای مینیمم است.**

c) **گراف G و عدد صحیح b داده شده است. با زمان خطی می‌توان تشخیص داد که مقدارِ درخت پوشای گلوگاه حداکثر b است.**

d) **با زمان خطی می‌توان درخت پوشای گلوگاه را بدست آورد.**

۱) ۱ ۲) ۲ ۳) ۳ ۴) ۴

۵۸- گراف غیرجهت‌دار وزن‌دار G داده شده است. فرض کنید S مجموعهٔ همهٔ درخت‌های پوشای کمینهٔ (MST) G است. گراف $G_S = (V_S, E_S)$ را به این صورت می‌سازیم که $V_S = S$ یعنی هر رأس G_S متناظر با یک MST از گراف G است یعنی اگر T_1 و T_2 دو MST از G باشند آنگاه T_1 و T_2 دو رأس در G_S هستند و همچنین یال (T_1, T_2) در صورتی که در G_S وجود دارد که درخت‌های پوشای T_1 و T_2 فقط در یک یال با هم اختلاف داشته باشند یعنی با برداشتن یک یال از T_1 و اضافه کردن یک یال دیگر به T_1 درخت T_2 حاصل شود. کدام گزینه در مورد G_S درست است؟

۱) G_S ناگزیر یک گراف کامل است.

۲) G_S لزوماً کامل نیست ولی حتماً هم‌بند است.

۳) G_S ممکن است همبند نباشد.

۴) G_S خود یک درخت است.

۵۹- گراف وزن‌دار غیرجهت‌دار $G = (V, E)$ را در نظر بگیرید. فرض کنید A و B یک افراز دلخواه از رأس‌های گراف است که $A \bigcup B = V$ و $A \bigcap B = \phi$. اگر یال $e = (u, v)$ کم‌وزن‌ترین یالی باشد که A را به B وصل می‌کند ($u \in A$ و $v \in B$)، آنگاه کدام گزاره درست است اگر:

الف) وزن یال‌های G متمایز باشد.

ب) در G یالهای با وزن یکسان وجود داشته باشد.

۱) (الف) یال e در همهٔ MSTهای G وجود دارد.

(ب) یال e در همهٔ MSTهای G وجود دارد.

۲) (الف) یال e حداقل در یک MST از G وجود دارد ولی نه لزوماً در همهٔ MST ها.
(ب) یال e حداقل در یک MST از G وجود دارد ولی نه لزوماً در همهٔ MST ها.

۳) (الف) یال e در همهٔ MST های G وجود دارد.
(ب) یال e حداقل در یک MST از G وجود دارد ولی نه لزوماً در همهٔ MST ها

۴) (الف) یال e حداقل در یک MST از G وجود دارد.
(ب) یال e ممکن است در هیچ MST از G وجود نداشته باشد.

۶۰- **گراف وزن‌دار غیرجهت‌دار $G=(V,E)$ را در نظر بگیرید که در آن وزن هر یال عددی بین صفر و یک است. وزن همه یالها را از یک کم می‌کنیم و گراف G' را می‌سازیم. حال با کراسکال، MST گراف G' را می‌یابیم و T' می‌نامیم آنگاه:**

۱) یالهای T' در گراف G تشکیل درخت پوشای کمینه می‌دهند.
۲) یالهای T' در گراف G تشکیل درخت پوشای بیشینه می‌دهند.
۳) یالهای T' برای گراف G ممکن است درخت پوشای کمینه یا درخت پوشای بیشینه تشکیل دهند.
۴) هیچ کدام

۶۱- **گراف ساده، همبند و غیرجهت‌دار $G=(V,E)$ با n رأس و e یال که وزن هر یال ۱ است را در نظر بگیرید. سریع‌ترین الگوریتم برای یافتن MST این گراف، از چه مرتبه‌ای است. فرض کنید گراف با لیست مجاورتی پیاده‌سازی شده است؟**

۱) $\theta(n)$ ۲) $\theta(e)$ ۳) $\theta(e.\lg n)$ ۴) $\theta(e+n.\lg n)$

۶۲- **می‌خواهیم در یک درخت آزاد $T=(V,E)$ با n رأس، اندازه کوچک‌ترین مجموعهٔ غالب T را بدست آوریم. زیرمجموعهٔ V' از رئوس را مجموعه غالب گویند هرگاه هر رأس که در V' نیست دستِ‌کم با یک رأس در V' مجاور باشد. سریع‌ترین الگوریتم کارا برای این کار از چه زمانی است؟**

۱) $\theta(n)$ ۲) $\theta(n^2)$
۳) $\theta(n.\lg n)$ ۴) راه حل چندجمله‌ای ندارد.

۶۳- **یک گراف «دو همبند» است اگر رأس برشی نداشته باشد. گراف هم‌بند و غیرجهت‌دار G با یال‌های مثبت را در نظر بگیرید. می‌خواهیم کم‌وزن‌ترین زیرگراف «دو هم بند» پوشای G را در صورت وجود بیابیم. یعنی زیرگرافی از G بیابیم که شامل همهٔ رئوس G باشد و رأس برشی نداشته باشد و جمع وزن‌های آن کمینه باشد. برای این کار الگوریتم زیر پیشنهاد**

می‌شود: «مانند الگوریتم کراسکال همهٔ یال‌ها را مرتب می‌کنیم. سپس تا زمانی که گراف «دو همبند» نشده است، یال‌ها را از کوچک به بزرگ اضافه می‌کنیم». کدام گزینه در بارهٔ این الگوریتم صحیح است؟

۱) همواره جواب درست را برمی‌گرداند.

۲) تنها روی درخت‌ها جواب درست را برمی‌گرداند.

۳) تنها روی گراف‌های غیر از درخت جواب درست را برمی‌گرداند.

۴) هیچکدام

۶۴- **کدام الگوریتم از نظر ساختار و پیاده‌سازی شبیه الگوریتم دایجسترا است؟**

۱) BFS ۲) DFS ۳) پریم ۴) گزینه‌های ۱ و ۳

۶۵- **گراف وزن‌دار و جهت‌دار G را با n رأس و e یال در نظر بگیرید. الگوریتم دایجسترا را یک بار برای یافتن وزن کوتاهترین مسیرها از رأس داده شدهٔ s به تک تک رأس‌ها فرا می‌خوانیم. اگر تعدادی از یال‌هایی که از رأس s خارج می‌شوند دارای وزن منفی و سایر یال‌ها دارای وزن مثبت باشند و دور منفی وجود نداشته باشد، کدام گزینه صحیح است؟**

۱) الگوریتم دایجسترا به خودی خود درست کار می‌کند.

۲) باید تغییری در گراف ایجاد کنیم و سپس دایجسترا را فراخوانی کنیم که در این صورت زمان اجرا $\theta(n)$ برابرِ زمان عادی است یعنی اگر زمان عادی x باشد، زمان اجرا $\theta(n.x)$ می‌باشد.

۳) باید تغییری در گراف ایجاد کنیم و سپس دایجسترا را فراخوانی کنیم که در این صورت زمان اجرا $\theta(n+e)$ برابر زمان عادی است.

۴) اصولاً نمی‌توان با دایجسترا، این مسئله را حل کرد.

۶۶- **گراف بدون جهت وزن‌دار G که وزن برخی یال‌های آن منفی است ولی دور منفی ندارد را در نظر بگیرید. اگر الگوریتم فلوید روی این گراف اجرا شود. کدام گزینه بهترین توصیف دربارهٔ خروجی الگوریتم است؟ فرض کنید در این الگوریتم، فاصله هر رأس تا خودش همواره صفر است و این مقدار تغییر نمی‌کند.**

۱) همیشه جواب درست را برمی‌گرداند.

۲) همیشه جواب نادرست را برمی‌گرداند.

۳) ممکن است جواب نادرست را برگرداند.

۴) در حلقه می‌افتد و به پایان نمی‌رسد.

۶۷- در الگوریتم بلمن فورد، اگر هنگام RELAX کردن یالها با انجام مقایسه، شاهد کاهش فاصله یک رأس نسبت به مبدأ باشیم، گوییم یک «کاهش» روی داده است. آیا گراف جهت‌دار و وزن‌دار و همبندی با n رأس و e یال وجود دارد که با الگوریتم بلمن فورد، تعداد کل کاهش‌ها در آن $\Omega(ne)$ شود؟

۱) برای گرافی که شامل وزن منفی است بله ولی برای گرافی که همه وزن‌ها مثبت است خیر.

۲) برای هر گرافی با هر وزنی خیر.

۳) برای گراف با هر وزنی بله.

۴) نمی‌توان اظهارنظر کرد.

۶۸- کدام یک از الگوریتم‌های زیر بر روی گراف از زمان خطی $\theta(n+e)$ نیست؟ $(|E|=e\,,|V|=n)$

۱) مرتب‌سازی توپولوژیکی رئوس یک گراف جهت‌دار در صورت وجود. اگر مرتب‌سازی توپولوژیکی وجود ندارد، تشخیص داده شود.

۲) یافتن کوتاهترین مسیرهای هم مبدأ در گراف بدون وزن یا در گراف وزن‌دار با وزن‌های مساوی.

۳) یافتن کوتاهترین مسیر وزن‌دار در گراف‌های بدون دور همبند.

۴) یافتن کوتاهترین مسیرهای وزن‌دار هم مبدأ در گراف جهت‌دار.

۶۹- گراف وزن‌دار و جهت‌دار G داده شده است. وزن یال $u \to v$ را به صورت $w(u,v)$ نمایش می‌دهیم. گراف G' را از روی G به این صورت می‌سازیم که $w(u,v)$ را با $w'(u,v)$ جایگزین می‌کنیم:

$$w'(u,v)=w(u,v)-\text{out}\deg(u)+\text{out}\deg(v)$$

در این صورت چند تا از گزاره‌های زیر صحیح هستند؟

a) **وزن یک دور در G با وزن همان دور در G' برابر است.**

b) **گراف G دور با وزن منفی دارد اگر و فقط اگر G' دور با وزن منفی داشته باشد.**

c) **حاصل جمع وزن یالهای این دو گراف با هم برابرند.**

d) **وزن کوتاهترین مسیر بین دو رأس در G و G' برابر است.**

e) **اگر جهت یال‌ها را نادیده بگیریم یعنی هر یال جهت‌دار را بدون جهت فرض کنیم، درخت فراگیر کمینهٔ (MST) در G و G' یکی است.**

۱) ۱ ۲) ۲ ۳) ۳ ۴) ۴

۷۰- چند تا از گزاره‌های زیر برای گراف $G=(V,E)$ با n رأس و e یال صحیح است؟

a) **در پیمایش عمقیِ یک DAG، هیچ گاه یال پشتی (پس سو، back) نداریم.**

b) **در پیمایش عمقیِ یک DAG، هیچ گاه یال ضربدری (چپ سو، cross) نداریم.**

c) **با داشتن تعداد یالها، در $O(n)$ و نه $O(n+e)$ می‌توان تشخیص داد که یک گراف ساده و غیرجهت‌دار، دور دارد یا خیر.**

d) **در $O(n+e)$ می‌توان تشخیص داد که یک گراف جهت‌دار، دور دارد یا خیر.**

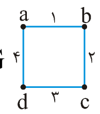
۱) ۱ ۲) ۲ ۳) ۳ ۴) ۴

۷۱- گراف p_n به صورت یک مسیر جهت‌دار با n رأس و $n-۱$ یال است. تعداد زیرگراف‌های p_n که آنها نیز به درستی به صورت یک مسیر (دست کم یک یالی) هستند چند تاست؟

۱) $۲^{n-۱}$ ۲) $n!$ ۳) $\frac{n(n-۱)}{۲}$ ۴) $\frac{n(n+۱)}{۲}$

[بخش دوم. پاسخنامه]

۱- گزینه (۳). a) درست است. یال (u,v) سبک‌ترین یال برش $(S, V-S)$ است و چون وزن‌ها متمایز هستند، این یال حتماً در MST باید انتخاب شود.

b) نادرست است. مثلاً در گراف  G کوتاهترین مسیر از a به d مسیر $a \to d$ است که وزن ۴ دارد، حال آنکه در MST این گراف اصلاً یال $\{a,d\}$ حضور ندارد.

c) درست است. مثلاً برای گراف  با شروع از a پیمایش عمقی اصلاً یال (a,b) را ملاقات نمی‌کند و این یال، cross می‌شود.

۲- گزینه (۱). فرض کنید V' ابتدا خالی است. با راس دلخواهی مثل $v \in V$ شروع کنید و v را به V' اضافه کنید و راس v و همه همسایگان راس v را از V حذف کنید. این پروسه را تا وقتی V خالی شود انجام دهید. مجموعه V' حاصل، مستقل و ماکسیمال است چون هر راس در $V-V'$ با راسی در V' مجاور است. الگوریتم از مرتبه $O(|V|+|E|)$ است چون هر راس و هر یال دقیقاً یکبار بررسی می‌شود.

توجه: در این سوال ما بدنبال مجموعه مستقل با بزرگترین (ماکزیمم) اندازه نیستیم. یافتن مجموعه مستقل ماکزیمم (نه ماکزیمال) یک مسئله $NP-complete$ است.

۳- گزینه (۱). رأسی مثل v که هنوز رنگ نشده را انتخاب کنید. v حداکثر با d راس مجاور است که ممکن است برخی از آنها رنگ شده باشند. حداکثر d رنگ از $d+1$ رنگ را نمی‌توان به راس v نسبت داد. (اگر همه رئوس مجاور v رنگ شده باشند و درجه راس v برابر d باشد، آنگاه d رنگ حذف می‌شوند) پس می‌توان راس v را رنگ کرد.این پروسه را تکرار کنید تا وقتی همه رئوس رنگ شوند. زمان این الگوریتم $O(|V|+|E|)$ است زیرا هر راس باید بررسی شود و هر یال حداکثر ۲ بار بررسی می‌شود.

۴- گزینه (۳). گراف جدید G' را بسازید به این صورت که به ازای هر راس $v \in V$ دو راس v_R (راس قرمز) و v_B (راس آبی) در G' ایجاد کنید. به ازای هر یال $(u,v) \in E$ اگر $color(u,v) = red$ آنگاه یال (u_R, v_R) و در غیر این صورت یال (u_B, v_B) را در G' ایجاد کنید. وزن یال ایجاد شده برابر همان وزن $\omega(u,v)$ است. در نهایت به ازای هر راس $v \in V$، یالهای (v_R, v_B) و (v_B, v_R) هر یک با وزن ۵ را در G' ایجاد کنید.حال بلمن فورد را دوبار اجرا کنید، یکبار با مبدا s_R و یکبار با مبدا s_B و بررسی کنید کدام یک از ۴ مسیر $s_R \to t_R$ ،$s_R \to t_B$ ،$s_B \to t_R$ ،$s_B \to t_B$ کوتاهترین است. ساخت G' از مرتبه

$O(n+e)$ است. G' شامل $2n$ راس و $n+e$ یال است و مرتبه بلمن فورد $O(n'e') = O(2n(n+e)) = O(n.e)$ است.

۵- گزینه (۲). الگوریتم پویایی ارائه می‌دهیم که تعداد مسیرهای از s به t را بیابد. مشخص کردن زوج یا فرد بودن این تعداد ساده است. ترتیب توپولوژیکی برای گراف فرض کنید $\langle v_1, v_2, ..., v_n \rangle$ است. $v_1 = s$ و $v_n = t$ فرض کنید. فرض کنید $S[i]$ تعداد مسیرهای از v_i به $v_n = t$ است می‌توان نوشت $S[i] = \sum_{j \in adj(i)} S[i]$ که $adj(i)$ مجموعه تمام رئوس v است که $(i,v) \in E$ یعنی رئوسی که از i به آنها یالی وجود دارد. شرط اولیه $S[n] = 1$ است.جواب مسئله $S[1]$ است. کل زمان حل تعداد $O(n)$ زیر مسئله برابر $O(e)$ است و ترتیب توپولوژیکی نیز از مرتبه $O(n+e)$ است. پس کل زمان $O(n+e)$ است.

۶- گزینه (۱). این الگوریتم را می‌توان به شکل تابع f پیاده‌سازی کرد:

```
f(E ,V)
1.sort edges in E by decreasing weight (e1,e2,...,e|E|)
2. for i ← 1 to |E|
   {
   if Reachable (G – ei ,ei1 ,ei2) is TRUE // هستند ei دوسر ei2 , ei1
   i ← i+1
   }
3. return G
```

G که در خط ۳ برمی‌گردد، MST است. زیر برنامه Reachable می‌تواند به شکل زیر پیاده شود:

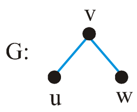
```
Reachable(G,a,b)
1. for(each vertex u ∈ V)
     {color[u]←WHITE, π[u]←NIL}
2. time ← 0
3. DFS – VISIT(a)
4. if b is BLACK return TRUE else return FALSE
```

الگوریتم DFS-VISIT را می‌توانید در کتاب طراحی الگوریتم پوران پژوهش ببینید. برای محاسبه زمان اجرای f، مرتب‌سازی در خط ۱، زمان $O(e.\lg e)$ می‌خواهد و زمان خط ۲

برابر $O(e.R(E,V))$ است که $R(E,V)$ زمان اجرای Reachable است برابر $O(e)$ است پس زمان کل $O(e.\lg e+e.e)=O(e^{۲})$ است.

توجه: پیاده‌سازی‌های سریعتری برای Reachable وجود دارد.

۷- گزینه (۴). (b نادرست است. به گراف مقابل دقت کنید:

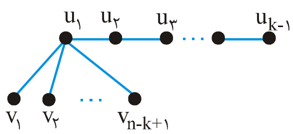


$$\mathrm{diam}(G)=\mathrm{dist}(u,w)=۲$$

$$\mathrm{apd}(G)=\frac{[\mathrm{dist}(u,v)+\mathrm{dist}(u,w)+\mathrm{dist}(v,w)]}{۳}=\frac{۴}{۳}$$

در واقع همیشه $\mathrm{diam}(G)\geq \mathrm{apd}(G)$ می‌باشد.

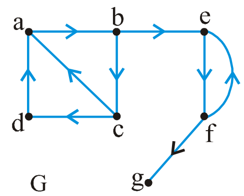
(a نادرست است. می‌توان نشان داد برای هر عدد طبیعی c، وجود دارد گراف G که $\frac{\mathrm{diam}(G)}{\mathrm{apd}(G)}>c$. گراف G را به شکل مقابل تصور کنید. این گراف n رأس دارد و قطر آن برابر $\mathrm{dist}(v_{۱},u_{k-۱})=k$ است. می‌توان $\mathrm{apd}(G)$ را یافت ولی ما یک کران بالا برای آن می‌یابیم. حداکثر تعداد kn مجموعه ۲ عضوی وجود دارد که حداقل یک عضو آن از $\{u_{۱},u_{۲},..,u_{k-۱}\}$ است. فاصله هر کدام از این جفت‌ها حداکثر k است. سایر جفت‌ها حداکثر با فاصله ۲ هستند. پس:



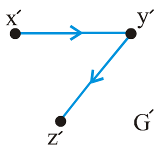
$$\mathrm{apd}(G)<\frac{۲\binom{n}{۲}+k^{۲}n}{\binom{n}{۲}}<۲+\frac{۲k^{۲}}{n-۱}$$

حال اگر $n-۱>۲k^{۲}$ انتخاب شود پس $\mathrm{apd}(G)<۳$ و با انتخاب $k>۳c$ داریم $\frac{\mathrm{diam}(G)}{\mathrm{apd}(G)}>\frac{۳c}{۳}=c$.

۸- گزینه (۳). برای درک سوال به گراف G و G' مقابل توجه کنید:



$$\rightarrow x'=\{abcd\}\ ,\ y'=\{ef\}\ ,\ z'=\{g\}\rightarrow$$

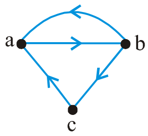


۹- گزینه (۱). (a) درست است ولی (b) غلط است. مثلاً در گراف $a\xrightarrow{۱}b\xrightarrow{۲}c\xrightarrow{۳}d$ اگر به ترتیب ab بعد bc و بعد cd، relax شوند کوتاهترین مسیر از a به d به دست می‌آید ولی

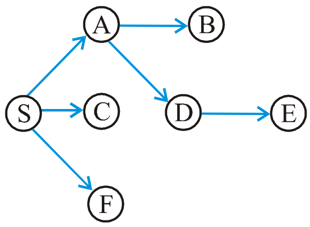
کوتاهترین مسیر از a به d شامل سه یال است. (c) غلط است زیرا مهم است که از چه رأسی شروع کرده‌ایم. در واقع برای یافتن قطر باید از یک رأس BFS اجرا کنیم و سپس از دورترین رأس مجدداً BFS اجرا کنیم آنگاه عمق درخت BFS حاصل، قطر گراف است.

۱۰- گزینه (۲). **توضیح گزینه ۱:** دایجسترا بعد از $|E|$ بار relax کردن و $|V|+|E|$ بار عملیات صف اولویت انجام دادن خاتمه می‌یابد ولی لزوماً جواب درست تولید نمی‌کند.

توضیح گزینه ۲: اگر وزن همه یالها k برابر شود، کوتاهترین مسیرها عوض نمی‌شود! مثلاً فرض کنید وزن همه یال‌ها برحسب کیلومتر داده شده‌اند حال می‌خواهیم بر حسب متر بیان کنیم یعنی باید در ۱۰۰۰ ضرب کنیم که کوتاه ترین مسیرها عوض نمی‌شود.

گزینه سوم غلط است. می‌توان چندین یال back داشت و فقط با حذف یک یال همه سیکل‌ها را از بین برد. به عنوان مثال، در گراف  دو سیکل وجود دارد: aba و abca. پیمایش عمقی با شروع از a دو یال back تولید می‌کند: (b,a) و (c,a). ولی حذف یال (a,b) از گراف هر دو سیکل را نابود و گراف را بدون سیکل می‌کند.

۱۱- گزینه (۲). به راحتی قابل بررسی است. درخت مسیر حاصل به شکل زیر است:



۱۲- گزینه (۳). کافیست وزن همه یالها را قرینه کنیم و سپس الگوریتم کوتاهترین مسیرهای هم مبدا در DAG را اجرا کنیم که از مرتبه $\theta(n+e)$ است. توجه کنید در حالت کلی $\theta(n+e)$ را نمی‌توان $\theta(e)$ یا $\theta(n)$ نوشت! در ضمن اگر گراف سیکل داشته باشد، این مسئله راه حل چند جمله‌ای ندارد.

۱۳- گزینه (۴). اگر گراف، جنگل باشد یعنی سیکل نداشته باشد، حال اگر یک یال به آن اضافه شود که یک سیکل ایجاد شود، از آن سیکل هر یالی که حذف شود، گراف بدون سیکل می‌شود. اگر گراف بیش از یک سیکل داشته باشد آنگاه یال decycling ندارد. زیرا حذف هر یالی گراف را بدون سیکل نمی‌کند. می‌دانیم DFS گراف غیر جهت‌دار فقط یال tree و back تولید می‌کند. پس اگر فقط یک یال back تولید کند یعنی گراف فقط یک سیکل

دارد که در نتیجه یال decycling دارد. در صورت وجود فقط یک یال back، آن یال و تمام یالهای tree که بین دو راس آن یال back هستند، یال decycling می‌باشند.

با تغییر کوچکی در DFS می‌توان یالهای decycling را یافت. وقتی اولین یال back مشاهده شد آن را ثبت کنیم؛ اگر دومین یال back مشاهده شد،کلا exit کنیم و NIL برگردانیم. اگر اصلاً یال back ندیدیم نیز NIL برگردانیم. در صورت مشاهده تنها یال پشتی (u,v)، می‌توان به انتهای DFS کد زیر را افزود و یالهای decycling را مشخص کرد:

$ans = (u.v)$

$x = u$

$while(x \neq v)$

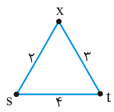
$add(x, \pi(x))$ to ans

$x = \pi(x)$

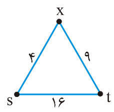
return ans

ظاهراً مرتبه DFS، $O(|V|+|E|)$ است. ولی ما بدنبال حداکثر ۲ یال back هستیم. یعنی اگر حداقل ۲ یال back وجود داشت یال decycling نداریم. پس مرتبه به $O(|V|)$ کاهش می‌یابد.

۱۴- گزینه (۲). **توضیح گزینه ۱ و ۲:** اگر وزن یالها را در یک عدد مثبت ضرب کنیم و یا با یک عدد جمع کنیم، MST تغییر نمی‌کند. اگر وزن یالها مثبت باشد و همه وزن‌ها را به توان ۲ برسانیم، MST تغییر نمی‌کند. اگر وزن یالها را با یک عدد جمع کنیم، ممکن است کوتاهترین مسیر بین دو راس عوض شود. اگر وزن یالها مثبت باشد و توان ۲ برسانیم، ممکن است کوتاهترین مسیر بین دو راس عوض شود. مثلاً در گراف



کوتاهترین مسیر از s به t مسیر $s \to t$ است. حال اگر وزنها به توان ۲ برسند، گراف



حاصل می‌شود که کوتاهترین مسیر از s به t مسیر $s \to x \to t$ است.

اگر وزن همه یالها در یک عدد مثبت ضرب شود، کوتاهترین مسیر بین دو راس، تغییر نمی‌کند.

توضیح گزینه ۳ و ۴: هر گراف غیر جهت‌دار همبند وزن‌دار، دارای MST است. یعنی وزن یالها هر چه باشد، مثبت یا منفی و حتی اگر سیکل منفی داشته باشیم، MST وجود دارد و کراسکال و پریم به درستی عمل می‌کنند.

۱۵- گزینه (۱). برای گراف داده شده خروجی فلوید عبارت است از:

$$\begin{array}{c|c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \left[\begin{array}{ccccc} ۰ & ۱ & ۳ & ۵ & ۷ \\ \infty & ۰ & ۲ & ۴ & ۶ \\ \infty & ۳ & ۰ & ۲ & ۴ \\ \infty & ۱ & ۳ & ۰ & ۷ \\ \infty & ۶ & ۸ & ۵ & ۰ \end{array}\right]
\end{array}$$

که کافیست در هر ستون ماکزیمم را بیابید.

۱۶- گزینه (۳). مرتبه همان مرتبه فلوید است.

۱۷- گزینه (۳). با خواندن لیست مجاورتی G و معکوس کردن آن می‌توان ترانهاده گراف را ایجاد کرد. از آنجا که تعداد عناصر لیست مجاورتی G برابر $|V|+|E|$ می‌باشد می‌توان این کار را در زمان $O(|V|+|E|)$ انجام داد.

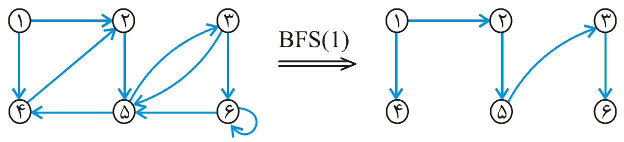
۱۸- گزینه (۲). پل یالی است که عضو هیچ سیکلی نیست. با پیمایشی DFS می‌توان سیکل‌ها را تشخیص داد. پس با این پیمایش می‌توان، پل‌ها را یافت که از مرتبه $O(|V|+|E|)$ است و چون گراف همبند است، $O(|V|+|E|)=O(|E|)$.

۱۹- گزینه (۴).

treeedge: ۱۲،۱۴،۲۵،۵۳،۳۶

backedge: ۳۵،۶۶،۶۵

crossedge: ۴۲،۵۴



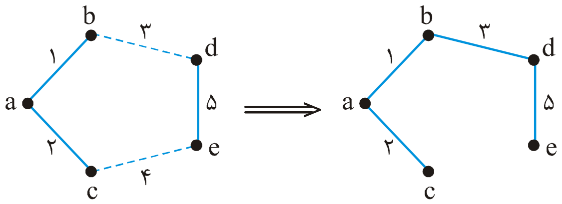
۲۰- گزینه (۳). بررسی کنید.

۲۱- گزینه (۱). می‌توان با BFS دوبخشی بودن را تشخیص داد (با DFS نیز می‌توان) به این صورت که اگر بتوان با ۲ رنگ گراف را رنگ کرد، دوبخشی است و برعکس. کافی است رأس شروع را رنگ ۱ بدهیم هر راسی که توسط این راس ملاقات می‌شود رنگ ۲ بدهیم و به همین ترتیب با BFS همه رئوس را رنگ دهیم و سپس یک بار یال‌ها را بررسی کنیم که آیا هر یال دو رأسش همرنگ هستند یا خیر، که در نهایت همان مرتبه BFS جواب است.

۲۲- گزینه (۴). درخت پوشای گلوگاه، درخت پوشایی است که سنگین‌ترین یالش از سنگین‌ترین یال سایر درخت‌های پوشا، سنگین‌تر نباشد! در گزینه ها،سنگین‌ترین یال وزن ۴ دارد و هیچ درخت پوشایی وجود ندارد که سنگین‌ترین یالش از ۴ کمتر باشد.

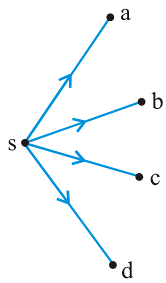
۲۳- گزینه (۱). فقط در صورتی که G خود درخت باشد، درخت DFS و درخت BFS یکسان هستند.

۲۴- گزینه (۴). مثلا گراف مقابل را در نظر بگیرید، $V_1=\{abc\}$, $V_2=\{de\}$ فرض کنید. آنگاه از یالهای bd ، ec که دو قسمت را به هم وصل می کند باید bd انتخاب شود، ولی حاصل MST نیست:



۲۵- گزینه (۱). یک روش این است که وزن همه یالها را قرینه کنیم و سپس کوتاهترین مسیر از s به همه رئوس را بیابیم. برای یافتن کوتاهترین مسیر، ترتیب توپولوژیکی را یافته و به همین ترتیب رئوس را برداشته و relax می‌کنیم. واضح است که مرتبه این عملیات با لیست مجاورتی، $\theta(n+e)$ است.

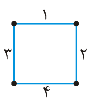
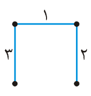
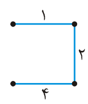
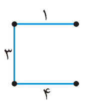
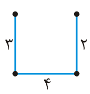
۲۶- گزینه (۲). **توضیح گزاره ۱:** نادرست. حافظه مصرفی BFS در واقع اندازه صف مورد استفاده در این الگوریتم است.اندازه صف از مرتبه $O(n)$ است. مثلاً در گراف DAG مقابل، حداکثر طول مسیر $L=1$ است ولی BFS بعد از ملاقات راس s، همه رئوس را وارد صف می‌کند و طول صف مورد نیاز $n-1$ است.



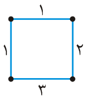
توضیح گزاره ۲: نادرست. DAG لزوماً گراف خلوت نیست و می‌تواند حداکثر $\frac{n(n-1)}{2}$ یال داشته باشد، پس مرتبه BFS و DFS روی DAG برابر $O(n+e)$ است و نه $O(n)$.

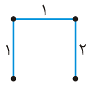
توضیح گزاره ۳: درست. منظور از $d[u]$ کوتاهترین فاصله راس s تا u است و مهم نیست که راس u در لیست مجاورتی کجاست. مثلاً در گراف فوق، فاصله راس s به همه رئوس a و b و c و d برابر ۱ است و مهم نیست در لیست مجاورتی s، کدام راس زودتر آمده است.

۲۷- گزینه (۱). با مرتبه $O(T)$، MST را می‌یابیم و سپس یال q را به MST می‌افزاییم در این صورت تعداد رئوس و یالهایش برابر n می‌باشد،حال از سیکل ایجاد شده سنگین‌ترین یال غیر از q را حذف می‌کنیم که از مرتبه $O(n)$ است. پس کل این عملیات از مرتبه $O(T+n)$ است.

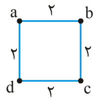
۲۸- گزینه (۱). خود MST زیباترین درخت فراگیر است. به عنوان مثال گراف  دارای ۴ درخت پوشای  ،  ،  ،  است که میزان زشتی

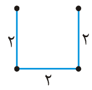
این ۴ درخت به ترتیب برابر ۳ و ۳ و ۴ و ۵ است که MST دارای کمترین میزان زشتی است البته ممکن است درخت‌های دیگری نیز با مینیمم زشتی باشند.

۲۹- گزینه (۲). فقط گزاره a صحیح است. اگر وزن همه یالها متمایز باشد آنگاه MST منحصر به فرد است ولی عکس این جمله صحیح نیست. مثلاً در گراف 

دو یال با وزن یکسان ۱ وجود دارد ولی این گراف فقط یک MST 

دارد.

۳۰- گزینه (۲). واضح است که $S_{G'} \subseteq S_G$ زیرا تعداد درخت‌های فراگیر کمینه G' از تعداد درخت‌های فراگیر کمینه G کمتر یا مساوی است. دقت کنید G' از حذف یال e در G، حاصل می‌شود به عنوان مثال گراف 

دارای ۴ درخت پوشای مینیمم است ولی اگر یال (a,b) را حذف کنیم گراف 

حاصل می‌شود که فقط یک درخت پوشای مینیمم دارد. بنابراین همیشه $S_{G'} \bigcap S_G = S_{G'}$ می‌باشد.

۳۱- گزینه (۱). حتماً رأس v وجود دارد و با BFS می‌توان این رأس را یافت. کافی است با شروع از رأس s، الگوریتم BFS را اجرا کنید. فرض کنید مجموعه رئوسی که در سطح اول توسط s ملاقات می‌شوند برابر L_1 باشد و مجموعه رئوسی که در سطح دوم ملاقات می‌شوند یعنی مجموعه رئوسی که فاصله‌شان تا s برابر ۲ است را برابر L_2 فرض کنیم و به همین ترتیب L_i برابر مجموعه رئوسی است که فاصله‌شان از s برابر i باشد. با توجه به صورت مسئله، رأس t در سطح L_d که $d > \frac{n}{2}$ ملاقات می‌شود. مشخص است که حداقل یکی از مجموعه‌های L_1 و L_2 و .. و L_{d-1} شامل یک رأس هستند، چون اگر هر کدام شامل حداقل ۲ رأس باشند، با توجه به اینکه $d > \frac{n}{2}$ آنگاه تعداد رئوس از n بیشتر می‌شود. پس وجود دارد L_i که شامل فقط یک رأس مثل v است و همین رأس v، رأسی است که اگر حذف شود آنگاه از s به t دیگر مسیری نخواهیم داشت.

۳۲- گزینه (۲). با مبدأ رأس v، الگوریتم BFS را اجرا می‌کنیم و مجموعه سطوح L_0, L_1, L_2 و ... را می‌یابیم که $L_0 = \{v\}$ و L_1 مجموعه رئوس با فاصله ۱ از v است. و به طور کلی L_i مجموعه رئوس با فاصله i از v است. فرض کنید $s(x)$ برابر تعداد کوتاهترین مسیرها از v به x است. برای هر رأس x در L_1 داریم $s(x) = 1$ چون تنها کوتاهترین مسیر برای رأس $x \in L_1$، یال $\{v,x\}$ است. برای سایر سطوح L_j که $j > 1$ اگر $y \in L_j$ آنگاه $s(y)$ برابر جمع $s(x)$ برای همه نودهای x در سطح L_{j-1} با یک یال به y است.

بعد از اجرای BFS، می‌توان تمام این مقادیر $s(x)$ را بـه ترتیـب بـرای $L_۱$ و $L_۲$ و $L_۳$ و ... محاسبه کرد. زمان محاسبه هر $s(x)$ حداکثر برابر درجه x است و از آنجـایی کـه جمـع کـل درجات برابر $۲e$ است پس زمان کل این الگوریتم $O(n+e)$ است.

۳۳- گزینه (۴). مجموعه یال‌هـای جهـت‌دار و غیرجهـت‌دار را E_{dir} و E_{undir} مـی‌نـامیم پـس $E = E_{dir} \cup E_{undir}$. با توجه بـه یـال‌هـای جهـت‌دار (یعنـی در گـراف $(G' = (V, E_{dir})$) ترتیب توپولوژیکی برای رئوس می‌یابیم که فرض کنید $(V_۱, V_۲, ..., V_n)$ باشد. دقـت کنیـد چون یال‌های جهت‌دار بدون سیکل هستند حتماً ترتیب توپولوژیکی وجـود دارد. جهـت هـر یال جهت‌دار از چپ به راست در ترتیب بدست آمده است یعنی از رأسی مثل V_i بـه رأسـی مثل V_j که $i < j$. حال کافی است به هـر یـال غیرجهـت‌دار مثـل (V_p, V_q) در E_{undir} جهتی نسبت دهیم که از $V_{min\{p,q\}}$ به $V_{max\{p,q\}}$ باشـد یعنـی جهـت‌هـا را از چـپ بـه راست نسبت دهیم. بنابراین اکنون گراف G کاملاً جهت‌دار و بدون سیکل اسـت. مرتبـه ایـن عملیات همان مرتبه ترتیب توپولوژیکی یعنی $O(n+e)$ است. زیرا نـسبت دادن جهـت بـه یال‌ها نیز از مرتبه خطی است.

۳۴- گزینه (۱). گراف $G' = (V', E')$ را از روی گراف G به صورت زیر مـی‌سـازیم: بـه ازای هـر رأس $v \in V(G)$ دو رأس v_r و v_s در G' ایجـاد مـی‌کنـیم. بـه ازای هـر یـال همـوار $(u,v) \in E(G)$، دو یال (u_r, v_r) و (u_s, v_s) در G' تولید می‌کنـیم. بـه ازای هـر یـال ناهموار $(u,v) \in E(G)$ یال‌های (u_s, v_r) و (v_s, u_r) را در G' ایجاد می‌کنـیم. اکنـون الگوریتم دایجسترا را برای یافتن کوتاهترین مسیر از s_s به t_s و از s_s به t_r اجرا می‌کنـیم و مینیمم این دو را انتخاب می‌کنیم. البته می‌توان یال‌های با وزن صفر از u_s به u_r به ازای هر $u \in V(G)$ ایجاد کنیم و دایجسترا را برای یافتن کوتـاهترین مـسیر از s_s بـه t_r اجـرا کنیم. در مورد زمان اجرا می‌توان گفت، تولید G' از مرتبه $O(n+e)$ اسـت و G' شـامل $۲n$ رأس و $۲e$ یال است. اجرای دایجـسترا از مرتبـه $O(۲e + ۲n.\lg n) = O(e + n.\lg n)$ بـا کمک هیپ فیبوناچی است. پس مرتبه کل همان مرتبه دایجسترا است.

۳۵- گزینه (۱). فرض کنید Adj نمایش لیست مجاورتی گراف جهت‌دار G است. درجـه خروجـی رأس u برابر است با طول $Adj[u]$، یعنی تعداد نودهای موجود در لیـست u را مـی‌شـماریم. پس محاسبه درجه خروجی همه رئوس از مرتبه $\theta(n+e)$ است. درجه ورودی رأس u برابر تعداد باری است که رأس u در همه لیست‌ها ظاهر شده است. اگر ما به ازای هر رأسـی همـه لیست‌ها را جستجو کنیم آنگاه مرتبه محاسبه درجه ورودی همه رئوس $\theta(n.e)$ اسـت. ولـی می‌توانیم یک آرایه $T[۱..n]$ تعریف کنیم و مقدار صفر به همه درایه‌های ایـن آرایـه بـدهیم. حال کافیست همه لیست‌ها را فقط یک بار جستجو کنـیم و هـر بـار رأس u را در لیـست‌هـا

دیدیم $T[u]$ را افزایش دهیم. در نهایت $T[i]$ نشان دهنده درجه ورودی رأس i است. مرتبه این الگوریتم $\theta(n+e)$ است و البته نیاز به $\theta(n)$ حافظه اضافی دارد.

۳۶- گزینه (۳). فرض کنید Adj لیست مجاورتی گراف G باشد و Adj۲ لیست مجاورتی گراف $G^۲$ باشد. کد زیر $G^۲$ را با لیست مجاورتی می‌یابد:

```
for each vertex v in Adj[u]
    for each vertex w in Adj[v]
        insert w in Adj۲[u].// edge (u,w) ∈ E^۲
```

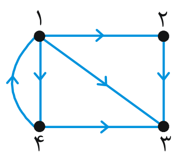
پس از محاسبه Adj۲ باید یال‌های تکراری که امکان دارد بوجود آید را از لیست‌ها حذف کنیم (ممکن است بین رأس u و w چنین مسیر به طول ۲ باشد). برای هر یال در Adj حداکثر n رأس بررسی می‌شوند، پس Adj۲ با مرتبه $O(n.e)$ بدست می‌آید. حذف یال‌های تکراری نیز از مرتبه $O(n+e)$ است پس کل الگوریتم از مرتبه $O(n.e)$ است. حال فرض کنید A ماتریس مجاورتی G باشد. ماتریس مجاورتی $G^۲$ عبارت است از $A^۲$ که می‌دانیم ضرب دو ماتریس $n\times n$ با روش استراسن از مرتبه $O(n^{\lg ۷})$ است.

۳۷- گزینه (۳). با ماتریس مجاورتی می‌توان با کمک دو حلقه for درایه $[i,j]$ را با $[j,i]$ تعویض کرد و مرتبه $\theta(n^۲)$ است. با لیست مجاورتی می‌توان کد زیر را نوشت:

```
allocate array G^T of size n
for (each v ∈ V)
    for (each u ∈ Adj[v] in G)
        insert v to Adj[u] in G^T
return G^T
```

در این کد به هر رأس و یال یکبار مراجعه می‌شود. پس مرتبه این کد $\theta(n+e)$ است.

۳۸- گزینه (۳). برای درک بهتر، در گراف مقابل رأس ۳ یک چاه فراگیر است زیرا درجه خروجی صفر و ورودی $n-۱=۳$ ($n=۴$ گراف $n=۴$ رأسی است) دارد.

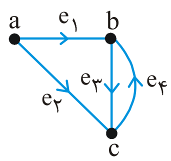


$$A=\begin{array}{c|cccc} & ۱ & ۲ & ۳ & ۴ \\ \hline ۱ & ۰ & ۱ & ۱ & ۱ \\ ۲ & ۰ & ۰ & ۱ & ۰ \\ ۳ & ۰ & ۰ & ۰ & ۰ \\ ۴ & ۱ & ۰ & ۱ & ۰ \end{array}$$

در واقع رأس k چاه فراگیر است اگر در سطر k همه درایه‌ها صفر باشند و در ستون k همه درایه‌ها (بجز درایه قطر) یک باشند. چاه فراگیر در صورت وجود در یک گراف منحصر به فرد است. الگوریتم به این صورت است:

سطر اول را از چپ بررسی می‌کنیم اگر همه درایه‌های این سطر صفر باشند آنگاه فقط رأس ۱ می‌تواند چاه فراگیر باشد حال ستون اول را بررسی می‌کنیم اگر همه درایه‌های این ستون ۱ باشند (بجز درایه قطر) آنگاه رأس ۱ چاه فراگیر است و در غیر این صورت، گراف چنین رأسی ندارد و الگوریتم خاتمه می‌یابد. حال فرض کنید در سطر ۱ در ستون k اولین درایه‌ی ۱ وجود دارد پس رئوس از ۱ تا $k-1$ نمی‌توانند چاه فراگیر باشند زیرا از رأس ۱ هیچ یالی به این رئوس وجود ندارد. پس ما $k-1$ رأس را با مرتبه $O(k)$ حذف کرده‌ایم. حال از بین رئوس باقیمانده اولین رأس را انتخاب می‌کنیم. فرض کنید این رأس j باشد. سطر j را شبیه سطر ۱ بررسی می‌کنیم، اگر همه درایه‌های سطر j صفر باشند آنگاه ستون j را بررسی می‌کنیم. در غیر این صورت اگر اولین درایه‌ی ۱ در ستون m باشد آنگاه رئوس تا $m-1$ و همچنین رأس j نمی‌توانند چاه فراگیر باشند. پس ما تاکنون $k+m$ رأس را با مرتبه $O(k+m)$ حذف کرده‌ایم. پس با مرتبه $O(n)$ می‌توان همه رئوس را حذف کرد و یا اینکه به رأس چاه فراگیر رسید.

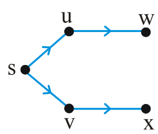
۳۹- گزینه (۲). به یک مثال توجه کنید.



$$B.B^T = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \end{array} \times \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \end{array} = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -2 \\ -1 & -2 & 3 \end{bmatrix} \end{array}$$

درایه‌های قطر اصلی، درجات رئوس گراف هستند (درجه یک رأس برابر مجموع درجات ورود و خروج آن رأس است) درایه‌های غیرقطر اصلی که همیشه منفی یا صفر هستند، قدرمطلق آنها تعداد یال بین دو رأس را نشان می‌دهد.

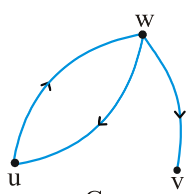
۴۰- گزینه (۲). یکی از درخت‌های کوتاهترین مسیر برای گراف گزینه ۲ به شکل



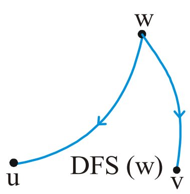
است ولی هیچ نوع پیمایش BFS با مبداء s این درخت را تولید نمی‌کند.

۴۱- گزینه (۳). با روش گزینه ۳ و با مرتبه $\theta(n)$ می‌توان قطر یک درخت را یافت.

۴۲- گزینه (۴). a و b هر دو غلط هستند. گراف G مقابل را در نظر بگیرید. پیمایش عمقی با شروع از رأس w نشان داده شده است. ابتدا w سپس u ملاقات شده و سپس u تمام شده و بعد v ملاقات شده و تمام شده و در نهایت



G

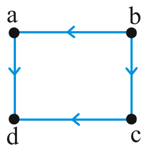


رأس w تمــام شــده اســت یعنــی $f[w]=۶$ و $f[v]=۵$ و $d[v]=۴$ و $f[u]=۳$ و $d[u]=۲$ و $d[w]=۱$

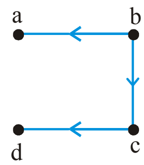
همانطور که می‌بینیم در گراف G از u به v مـسیری وجـود دارد و همچنـین $d[u]<d[v]$ ولی در جنگل حاصل از پیمـایش عمقـی، رأس v نـواده رأس u نیـست (a غلـط). همچنـین $d[v]>f[u]$ پس b نیز غلط است.

۴۳- گزینه (۴). الگوریتم زیر با مرتبـه $O(n.e)$ تـشخیص مـی‌دهـد گـراف جهت‌دار G، متـصل یکطرفه است: برای هر رأس $u\in V$، تابع $DFS-VISIT(u)$ را اجرا می‌کنیم. توجه کنیـد فقط همین تابع برای هر رأس u اجرا می‌شود یعنی از رأس u شروع می‌کنیم و هر رأسی کـه بتوانیم را ملاقات می‌کنیم و بررسی نمی‌کنیم آیا رأس ملاقات نشده‌ای بـاقی مانـده اسـت یـا خیر. اگر یال Cross یا forward وجود داشته باشد آنگاه گراف متـصل یکطرفـه نیـست، در غیر این صورت گراف متصل یکطرفه است.

به عنوان مثال گراف جهت‌دار مقابل را در نظر بگیرید:



به ازای هر رأس تابع DFS-VISIT را اجرا می‌کنیم:

DFS-VISIT (b)


DFS-VISIT (a)


DFS-VISIT (d)


DFS-VISIT (c)


همانطور که مشاهده می‌کنید، در DFS-VISIT(b) یال (a,d) یـال cross اسـت. پـس ایـن گراف متصل یکطرفه نیست (از b به d دو مسیر $b\rightarrow c\rightarrow d$ و $b\rightarrow a\rightarrow d$ وجود دارد)

۴۴- گزینه (۳). روی گراف G ترتیب توپولوژیکی را با مرتبه $O(n+e)$ می‌یابیم. تمام مـسیرهای بین s و t فقط از رئوس بین s و t در ترتیب توپولوژیکی استفاده می‌کنند.

حال می‌توان با برنامه‌نویسی پویا تعداد مسیرهای از s به t را شمارش کرد. برای ایـن منظـور به رأس s برچسب ۱ را نسبت می‌دهیم (از s به s یک مسیر وجـود دارد) و بـه سـایر رئـوس برچسب صفر نسبت می‌دهیم. سپس به ترتیب هر نود مثل i را از ترتیب توپولوژیکی انتخـاب

کرده و با رابطه $paths[i] = \sum_{(j,i)\in E} paths[j]$ تعداد مسیرهای از s به i را می‌شماریم. در نهایت $paths[t]$ جواب است.

برای درک بهتر، فرض کنید می‌خواهیم در گراف داده شده در صورت سؤال، تعداد مسیرهای از p به v را بشماریم. ترتیب توپولوژیکی این گراف می‌تواند به شکل «p n o s m r y v x w z u q t» باشد.

حال از رأس p تا رأس v مقدار paths را محاسبه می‌کنیم:

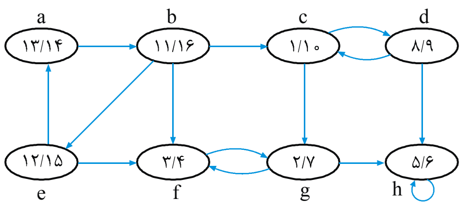
$paths[p] = ۱$ و $paths[n] = ۰$ و $paths[o] = paths[p] + paths[n] = ۱$ و

$paths[s] = paths[p] + paths[o] = ۲$ و $paths[m] = ۰$ و

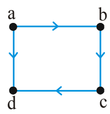
$paths[r] = paths[o] + paths[s] + paths[m] = ۳$ و $paths[y] = paths[r] = ۳$

$paths[v] = paths[o] + paths[y] = ۴$

۴۵- گزینه (۱). گراف بدون سیکل است اگر و فقط اگر پیمایش DFS یال back تولید نکند. پس DFS را اجرا کنیم اگر اصلاً یال back دیده نشود به این معنی است که همه یال‌ها tree هستند و حداکثر $n-۱$ یال tree وجود دارد که مرتبه $O(n)$ است. و اگر یال back دیده شود الگوریتم را خاتمه دهیم که در این صورت حداکثر n یال بررسی شده‌اند که در این صورت نیز مرتبه $O(n)$ است.

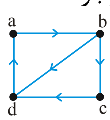


۴۶- گزینه (۲). a) نادرست است. مثلاً برای گراف مقابل پیمایش عمقی را با یک ترتیب دلخواه انجام داده‌ایم و داخل هر نود مقدار d / f (discovery/finish) را ثبت کرده‌ایم حال اگر مجدد روی همین گراف پیمایش عمقی را به ترتیب صعودی fها انجام دهیم، با اولین فراخوانیِ DFS–VISIT، {fgh} ملاقات می‌شوند که این یک مولفه متصل قوی نیست. b) درست است. مثلاً گراف



دارای ۴ مولفهٔ متصل قوی {a} و {b} و {c} و {d} است. حال اگر یال (c, a) به گراف اضافه شود آنگاه ۲ مولفه متصل قوی {abc} و {d} خواهیم داشت.

۴۷- گزینه (۱). گراف متصل قوی گراف جهت‌داری است که به ازای هر دو رأس x و y هم از x به y و هم از y به x حداقل یک مسیر وجود داشته باشد یعنی از هر رأس به همه رئوس دسترسی وجود داشته باشد. مثلاً گراف

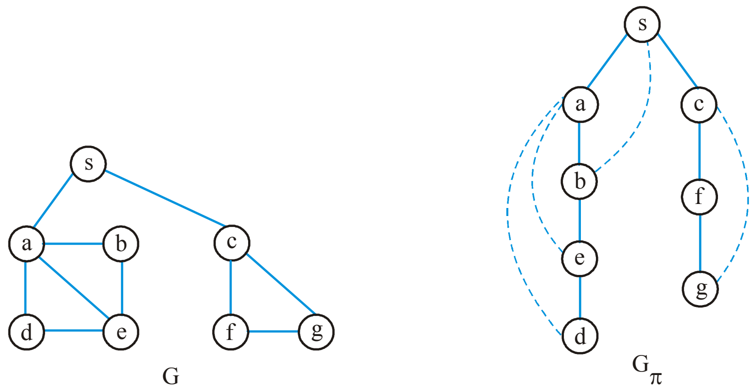


متصل قوی است ولی تور اویلری ندارد

(عبارت c غلط است). به طور کلی شرط لازم و کافی برای وجود تور اویلری در گراف غیرجهت‌دار آن است که گراف همبند باشد و درجه همه رئوس زوج باشد. شرط لازم و کافی برای وجود تور اویلری در گراف جهت‌دار آن است که گراف همبند باشد و درجه ورودی هر رأس با درجه خروجی آن مساوی باشد. چون صورت سؤال گفته گراف G متصل قوی است پس عبارت a صحیح است و نیازی به ذکر کلمه همبند نیست. مرتبه یافتن تور اویلری به طور کلی $O(n+e)$ است ولی چون گراف همبند است و $n \in O(e)$ پس مرتبه یافتن تور اویلری $O(e)$ است. برای یافتن تور اویلری، از یک راس دلخواه شروع کنید و یال‌ها را طی کنید طوری که هر یال حداکثر یک بار طی شود، تا وقتی که به رأس شروع باز گردید. اکنون شما یک سیکل مثل C_0 یافته‌اید، حال یال‌های C_0 را از گراف G حذف کنید و نام گراف جدید را G_1 بگذارید. در گراف G_1 نیز درجه ورودی هر رأس با خروجی آن مساوی است. هر جزء همبند در G_1 حتماً حداقل یک رأس در C_0 دارد (در غیر این صورت گراف اصلی G ناهمبند است). اکنون برای هر جزء همبند G_1، به طور بازگشتی یک تور اویلری بیابید. و در نهایت همه این تورها را با C_0 ادغام کنید.

۴۸- گزینه (۴). همه عبارات صحیح هستند.

a) برای درک بهتر، گراف G مقابل و درخت حاصل از پیمایش عمقی G_π با شروع از s را در نظر بگیرید. ریشه G_π رأس s است که دو فرزند دارد و همانطور که در G مشاهده می‌کنید، رأس s انفصال است.



برای اثبات فرض کنید s ریشه G_π است و حداقل دو فرزند دارد می‌دانیم پیمایش عمقی گراف غیرجهت‌دار یال cross ندارد پس هیچ یالی از زیر درخت چپ s به زیر درخت راست s وجود ندارد بنابراین اگر s از گراف G حذف شود حتماً گراف ناهمبند می‌شود.

b) فرض کنید v یک رأس غیرریشه در G_π باشد. دو حالت را در نظر می‌گیریم. حالت ۱: اگر v یک فرزند مثل s طبق صورت مسئله داشته باشد آنگاه حذف v باعث می‌شود s از پدر v جدا شود پس v یک نقطه انفصال است. حالت ۲: اگر رأسی مثل s وجود نداشته باشد آنگاه نشان می‌دهیم رأس v نقطه انفصال نیست. برای اثبات فرض کنید $v_۰$ پدر v باشد و $v_۱,...,v_m$ فرزندان v باشند. آنگاه طبق فرض، مسیری از v_i به $v_۰$ به ازای هر $i=۱,...,m$ وجود دارد. یعنی مجموعه $S=\{v_۰,v_۱,...,v_m\}$ همگی در یک مولفه همبند هستند. پس حذف v، این رئوس را از هم جدا نمی‌کند. واضح است که هر مسیری که از v می‌گذرد باید از دو رأس S بگذرد. پس اگر مسیری از رئوس u و u' در گراف اولیه G باشد، آنگاه هنوز مسیری از u به u' در گراف پس از حذف v وجود خواهد داشت.

c) می‌توان در تابع DFS-VISIT دو دستور اضافه کرده و مقدار low[u] را برای هر رأس u محاسبه کرد. (برای بررسی تابع DFS-VISIT به کتاب طراحی الگوریتم پوران پژوهش مراجعه کنید)

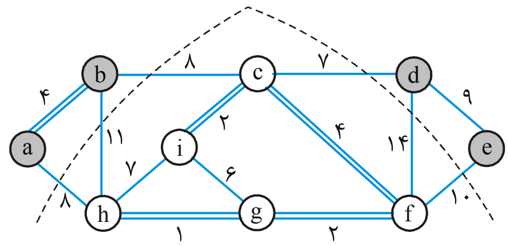
```
DFS − VISIT(G,u)
۱ time = time + ۱
۲ d[u] = time
۳ color[u] = GRAY
۴ low[u] = d[u]
۵ for(each v ∈ Adj[u])
۶    if (color [v] == WHITE)
       {
۷        π[v] = u
۸         DFS − VISIT(G, v)
۹         low[u] = min {low[u], low[v]}
       }
۱۰ Color[u] = BLACK
۱۱ time = time + ۱
۱۲ f[u] = time
```

دستورات ۴ و ۹ به کد اصلی اضافه شده‌اند. پس مرتبه محاسبه low برای همه رئوس همان مرتبه پیمایش عمقی یعنی $O(n+e)$ است و چون گراف همبند است مرتبه $O(e)$ است.

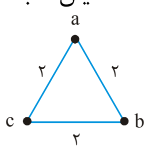
d) برای یافتن نقاط انفصال، الگوریتم DFS-VISIT را با شروع از رأس دلخواه $v_۰$ اجرا می‌کنیم. در این کد پس از دستور $Color[u]=BLACK$، بررسی می‌کنیم اگر $low[u]=d[u]$ آنگاه u را به عنوان نقطه انفصال اعلام می‌کنیم. همچنین خود $v_۰$ اگر بیش از یک فرزند در درخت حاصل داشته باشد نیز به عنوان نقطه انفصال اعلام می‌شود.

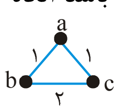
درستی سایر گزاره‌ها را هم بررسی کنید.

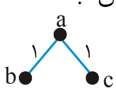
۴۹- گزینه (۲). برای درک بهتر سوال، گراف G و برش $(S, V-S)$ که $S=\{a,b,d,e\}$ و $V-S=\{c,f,g,h,i\}$ در شکل مشخص شده‌اند.



زیرمجموعه $A=\{ab,ci,cf,fg,gh\}$ از یال‌ها در شکل مشخص شده است. این زیرمجموعه در واقع زیرمجموعه‌ای از MST گراف است و همانطور که مشاهده می‌کنید برش $(S, V-S)$ به A احترام می‌گذارد و هیچ یال A را قطع نمی‌کند. یال‌هایی که از این برش عبور می‌کنند (برش را قطع می‌کنند) عبارتند از ah، bh، bc، cd، df و ef که یال cd یال سبک برای این برش است و این یال برای A امن است زیرا اگر به A اضافه شود، A همچنان زیرمجموعه‌ای از MST گراف خواهد بود. یال bc یا ah نیز برای A امن هستند چون اگر مثلاً یال bc به A اضافه شود، A همچنان زیرمجموعه‌ای از MST باقی می‌ماند ولی یال bc برای برش گفته شده، یال سبک نیست یعنی گزاره (b) نادرست است.

گزاره (c) درست است چون همه یال‌های موجود در MST حتماً یال سبک عبور کننده از حداقل یک برش هستند گزاره (d) غلط است مثلاً در گراف 

هر برشی که در نظر بگیرید، ۲ تا از یال‌ها، یال سبک برای آن هستند مثلاً برای برش $S=\{a\}$ و $V-S=\{b,c\}$، هر دو یال ab و ac یال سبک هستند و یا برای برش $S=\{b\}$ و $V-S=\{a,c\}$، هر دو یال ba و bc یال سبک هستند ولی اگر این یال‌های سبک را در یک مجموعه قرار دهیم مجموعه $\{ab,ac,bc\}$ حاصل می‌شود که MST نیست. گزاره (e) غلط است. اگر برای هر برش، یال سبک منحصر به فردی عبور کننده از آن برش وجود داشته باشد آنگاه MST منحصر به فرد است ولی عکس این جمله صحیح نیست. مثلاً برای گراف 

، MST منحصر به فرد است و به شکل 

است، حال آنکه برای برش $S=\{a\}$ و $V-S=\{b,c\}$ هر دو یال ac و ab یال سبک هستند یعنی سبک عبورکننده از این برش، منحصر به فرد نیست.