نمونه سوالات مربوط به بخش ۸.۵

۸.۵.۱ گراف بدون جهت و وزن دار $G=(V,E)$ را در نظر بگیرید که در آن وزن هر یال عددی بین صفر و یک است. از G گراف G' را با همان راس ها و یال ها می سازیم و وزن هر یال e در G' را برابر $1 - w_e$ قرار می دهیم. حال با استفاده از الگوریتم کروسکال درخت فراگیر کمینه ی G' را (به نام T') می سازیم.

کدام یک از روابط زیر درست است:

1) **یال های T' در گراف G تشکیل درخت پوشا بیشینه میدهد.**
2) **یال های T' در گراف G تشکیل MST میدهد.**
3) **هر MST در G ، دست کم در یک یال با T' اشتراک دارد.**
4) **هیچ کدام**

راه حل:

گزینه ۱: اگر بخواهیم درخت فراگیر بیشینه را در G با الگوریتم کروسکال بسازیم، باید یالها را به ترتیب نزولی مرتب کنیم. این ترتیب مشابه ترتیب صعودی است اگر یال ها وزنشان $1-w_e$ شده باشد.

۸.۵.۲ قدرت یک درخت فراگیر در G برابر وزن سنگین ترین یال در آن درخت است. می خواهیم در گراف داده شده G از میان تمام درخت های فراگیر، آن را انتخاب کنیم که قدرتش از همه کمتر باشد.خروجی کدام یک از الگوریتم های پریم یا کروسکال همواره کم قدرت ترین درخت فراگیر است؟

راه حل:

کم قدرت ترین بودن الگوریتم کروسکال معلوم می باشد. این الگوریتم به صورت حریصانه و بعد از مرتب سازی یال ها بر اساس وزنشان هر یالی را که میبیند برای زیر درخت فراگیر انتخاب میکند تا هرچه زودتر درخت فراگیر کمینه اش را بسازد و زمان کمتر به معنی استفاده از یال های سبکتر است.

در الگوریتم پریم اگر یال های کم قدرت ترین درخت فراگیر را در نظر بگیریم میبینیم که این الگوریتم تمامی این یال ها را مشاهده می‌کند و در پایان کار جوابش بهتر یا مساوی جواب بهینه است.

در حالت کلی اثبات میشود که دنباله های وزنی یال های تمام درخت های فراگیر کمینه ی یک گراف ثابت با هم برابرند.

۸.۵.۳ فرض کنید G یک گراف ساده ، بدون جهت و وزن دار با وزن های مثبت است و K تعداد درخت های فراگیر کمینه ی مختلف G است. دو درخت فراگیر کمینه مختلف اند اگر دست کم در یک یال اختلاف داشته باشند.کدام ۲ گزاره زیر در مورد رابطه K با متفاوت بودن وزن یال های G درست اند؟

۱)اگر k=1، ناگزیر وزن یال های G دو به دو متفاوت است.

۲)اگر دست کم دوتا از یال های G وزن یکسان داشته باشند، انگاه k>=2

راه حل:

۱)فرض کنید G یک درخت ساده و همبند با ۱- n یال باشد که وزن همه یال های آنها یک است. روشن است که G تنها یک درخت فراگیر کمینه دارد و آن هم خودش است و در این حالت k=1 ولی وزن یال ها همگی یکسان اند پس این جمله نادرست است.

۲) همان مثال گزاره یک را در نظر بگیرید، این جمله با توجه به همان مثال نادرست است.

۸.۵.۴ در گراف وزن دار G فرض کنید که وزن های یال ها دو به دو متفاوت اند و مجموعه های A,B یک افراز دلخواه از راس های گراف اند که A ∪ B =V . اگر یال e کم وزن ترین یالی باشد که A را به B وصل میکند، آنگاه کدام یک از گزاره های زیر درست اند؟

۱)e ناگزیر در تمام درخت های فراگیر کمینه G آمده است.

۲)e در هیچ یک از درخت های فراگیر کمینه G نیامده است.

۳)e شاید در بعضی درخت های پوشا کمینه G بیاید.

۴) هیچ کدام

راه حل:

گزینه یک: فرض خلف: e در درخت فراگیر کمینه نیامده باشد، پس بین u , v یک مسیر در درخت فراگیر مربوطه وجود دارد فرض کنیم این میسر $u = a_0 , a_1 , a_2 , \dots , a_{k-1} , a_k = v$ باشد. روی مسیر از u شروع میکنیم و به v میرویم. روشن است که اندیس i وجود دارد که $a_i \in A$ و $a_{i+1} \in B$. حال چون e کم وزن ترین بال بین مجموعه A و B است، پس وزن یال $e' = (a_i + a_{i+1})$ حتما از e بیشتر است. پس اکنون با افزودن e و حذف e' میتوان به یک درخت فراگیر کمینه با وزن کمتر رسید که تناقض است پس e حتما در درخت پوشا کمینه هست.

۸.۵.۵ گراف ساده و وزن دار G را در نظر بگیرید. کوتاهترین مسیر بین هر دو راس u , v را محاسبه و خود مسیر را در درایه D_{uv} از ماتریس D ذخیره می کنیم، به وزن هر یال G به طور دقیق به اندازه یک واحد افزایش می دهیم. پاسخ دو پرسش زیر کدام است؟

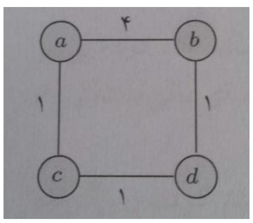
الف)آیا برای دو راس u , v مسیر ذخیره شده در D_{uv} هم‌چنان کم وزن ترین مسیر است؟

ب)آیا هر درخت فراگیر کمینه در G همچنان درخت فراگیر کمینه باقی می ماند؟

راه حل:

الف: خیر – به عنوان مثال گراف زیر را در نظر بگیرید، در این گراف کوتاهترین مسیر بین راس های a,b وزنی برابر ۳ دارد

<a,c,d,b> حال آنکه با افزایش یک واحد مسیر بهینه بین a,b به ۵ تغییر کرده است.



شکل شماره ۱

ب: بله- چون ترتیب یال ها از نظر وزن ثابت می ماند و الگوریتم کروسکال همان رویه پیشین را می پیماید.

۸.۵.۶ T درخت فراگیر کمینه برای گراف G است، وزن یک یال e را کاهش می دهیم. با چه مرتبه ای می توان درخت فراگیر کمینه گراف جدید را به دست آورد؟

راه حل:

در صورتی که این یال در T باشد، T همچنان درخت فراگیر کمینه باقی می ماند. اما در صورتی که این یال در T نباشد، تنها تغییر ممکن افزودن e به T و حذف یک یال از T است.می دانیم در صورت افزودن e به T، یک دور به وجود می آید و سنگین ترین یال این دور باید حذف شود.

برای یافتن این دور کافی است مسیر یکتای بین دو سر e در T را بیابیم و از آنجا که T درخت است، این کار با الگوریتم BFS یا DFS از زمان $O(|V(T)|) = O(|E(T)|)$ قابل انجام است.

۸.۵.۷ درخت فراگیر گلوگاه T در یک گراف بدون جهت و وزندار G یک درخت فراگیر است که وزن سنگین ترین یال آن نسبت به هر درخت فراگیر دیگر کمینه باشد. می گوییم که مقدار یک درخت فراگیر گلوگاه ، وزن سنگین ترین یال آن درخت است. کدام یک از گزینه های زیر درست است؟

۱) هر درخت فراگیر کمینه یک درخت فراگیر گلوگاه است.

۲)هر درخت فراگیر گلوگاه یک درخت فراگیر کمینه است.

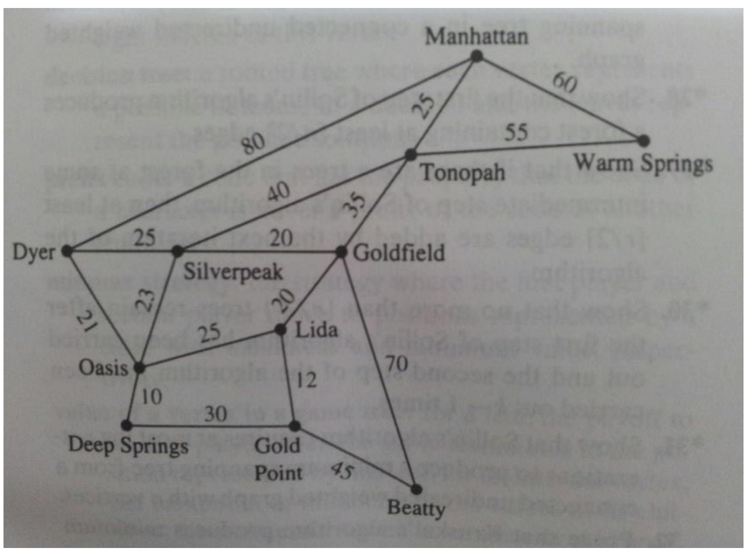
۳)هر درخت فراگیر بیشینه یک درخت فراگیر گلوگاه است.

۴)هر درخت فراگیر گلوگاه یک درخت فراگیر بیشینه است.

راه حل:

گزینه یک : فرض کنید T که یک درخت فراگیر کمینه است و گلوگاه نیست، پس یک درخت فراگیر به نام T' وجود دارد که وزن سنگین ترین یال T به نام e از همه یال های T' بیشتر است. یال e را از T برمیداریم و بجای آن یک یال از T' قرار می دهیم که درخت ساخته شود.درختی را به دست آوردیم که کمینه تر از T است. تناقض!

۸.۵.۸ درخت پوشا کمینه گراف زیر را بیابید.



شکل شماره۲

راه حل:

Deep Springs- Oasis , Oasis-Dyer , Oasis-Silverpeak , Silverpeak-Goldfield , Lida-Gold Piont , Gold Piont-Beatty , Lida-Goldfield , Goldfield- Tonopah , Tonopah-Manhattan , Tonopah- Warm Springs.

۸.۵.۹ گراف ساده و همبند و بدون جهت G را در تظر بگیرید که وزن تمامی یال های آن یک است،سریع ترین الگوریتم برای یافتن درخت پوشا کمینه از چه مرتبه زمانی است؟

راه حل:

از آنجا که هر درخت فراگیری در این گراف کمینه است پس کافیست با استفاده از الگوریتم BFS یا DFS یک درخت پوشا کمینه در آن بیابیم.

۸.۵.۱۰ نشان دهید که درخت پوشا کمینه در گراف همبند وزندار که وزن تمامی یال ها متفاوت است یکتاست.

راه حل:

اثبات با استفاده از برهان خلف:

فرض می کنیم دو درخت پوشا کمینه وجود دارد A,B

فرض میکنیم یالی با کمترین وزن(e) فقط در یکی از این دو وجود دارد(درختA). پس درخت B شامل e نمیباشد.

پس **B** U{e} تشکیل دور میدهد که یکی دیگر از یال های این دور (e') در درخت A وجود ندارد. با توجه با اینکه وزن یال e از یال e' کمتر است ($W(e) < W(e')$) و T=B U {e}\{e'} یک درخت پوشا کمینه است در این صورت وزن درخت T از وزن B کمتر است. تناقض!

۸.۵.۱۱ الگوریتم زیر جهت به دست آوردن درخت فراگیر کمینه M در هر گراف ساده و وزندار و بدون جهت G داده شده است.

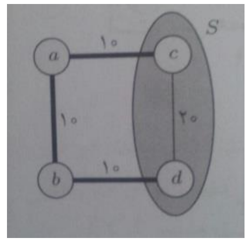
۱.در صورتی که گراف دست کم دو راس دارد, راس های آن را به دو مجموعه دلخواه s , v-s تقسیم کن.

۲.درخت فراگیر کمینه را برای هر دو مجموعه و به صورت بازگشتی توسز همین الگوریتم به دست آور.

۳.کم وزن ترین یال بین s , v-s از گام اول را به M اضافه کن.

آیا این الگوریتم همواره درست کار می کند؟

راه حل:



شکل شماره۳

دقت کنید که در جواب بهینه ممکن است زیرگراف القایی هر یک از دو بخش s , v-s ناگزیر خود یک زیردرخت فراگیر کمینه نباشد.

مثال: در شکل بالا برای رسیدن به جواب بهینه می بایست از مجموعه s تنها دو راس(و هیچ یال) انتخاب شود که با الگوریتم گفته شده در تناقض است.

۸.۵.۱۲ در مورد درخت فراگیر کمینه در گراف بدون جهت و وزندار کدام گزینه بهترین جواب است؟

۱ . الکوریتم های پریم و کروسکال ممکن است درخت های بهینه مختلفی را محاسبه کنند.

۲. اگر تعدلد یال های گراف خیلی زیاد باشند الگوریتم پریم ناگزیر سریع تر از کروسکال است.

۳. اگر یال ها منفی ولی گراف بدون دور باشد هر دو الگوریتم نادرست جواب می دهند.

۴. در الگوریتم پریم درخت فراگیر کمینه همان درخت جستوجو سطح اول است.

راه حل:

گزینه یک: گراف کامل n راسی که وزن تمام یال های آن ۱ است را در نظر بگیرید. هر درخت n-1 راسی در این گراف می تواند جواب باشد الگوریتم پریم چنین گرافی را بر اساس اندیس راس ها اضافه می کند در حالی که در الگوریتم کروسکال ترتیب یال دهی ها مهم است. پس ممکن است متفاوت باشند.