نمونه سوالات مربوط به بخش 8.6

8.6.1 حسن تعدادی چراگاه دارد که بین بعضی از آنها یالهای وزن‌دار است. هر چراگاه متعلق به یک گاو است و همه‌ی گاوها می‌خواهد در یک چراگاه جمع شوند. چراگاهی را پیدا کنید که گاوها در آن جمع شوند و جمع مسافت طی شده توسط آنها کمینه شود.

راه حل:

مسأله را برعکس حل کنید. فرض کنید چراگاه محل تجمع را دارید و می‌خواهید گاوها به جای اول خود برگردند. این مسأله‌ی کوتاه‌ترین مسیر است. برای حل سوال کوتاهترین مسیر را از هر رأس حساب کنید و به ازای مکان‌های گاوها با هم جمع کنید. رأس با کمترین مجموع جواب است.

**8.6.2 حسن می‌خواهد به یک مزرعه‌ی دیگر نقل مکان کند که مسافتی که هر روز طی می‌کند را کمینه کند. جایی که حسن به آن می‌رود N $(1\leq N\leq 10000)$ شهر دارد. M جاده‌ی دوطرفه $(1\leq M\leq 50000)$ این شهرها را به هم وصل کرده‌اند. همه‌ی شهرها از یکدیگر با دنباله‌ای از جاده‌ها قابل دسترس هستند. حسن باید بهترین شهر را به عنوان خانه‌ی جدیدش انتخاب کند. Kفروشگاه در شهرها هستند $(1\leq K\leq 5)$ که حسن باید هر روز به آنها برود. به علاوه هر روز حسن می‌خواهد از خانه‌اش به این K شهر برود و به خانه‌اش برگردد. حسن با هر ترتیب دلخواه می‌تواند فروشگاه‌ها را ببیند. وقتی یک شهر را به عنوان خانه (محل ساخت مزرعه جدید) انتخاب می‌کند، می‌خواهد فقط خانه‌اش از بین $N-K$ شهری باشد که فروشگاه ندارند ، چون قیمت خانه در این شهرها کمتر است.
به حسن کمک کنید که کمترین فاصله‌ای که هر روز باید طی کند را کمینه کند؛ اگر بخواهد مزرعه‌ی جدیدش را در بهترین مکان بسازد و ترتیب دیدن مغازه‌ها را هم تا حد امکان هوشمندانه انتخاب کند.**

راه حل:

کوتاهترین مسیر را از این Kشهر پیدا می‌کنیم. به ازای هر شهر از $N-K$ شهر به عنوان خانه جدید و دو شهر از Kشهر فاصله‌ی این دو شهر از خانه جدید و مسافت کمترین ترتیب Kشهر طوری که این دو شهر دو سر گشت قرار بگیرند؛ را حساب می‌کنیم. کمینه‌ی اینها جواب مسأله است. (به ازای هر ترتیب Kشهر باید کوتاهترین مسیر را حساب کنیم. می‌توانیم همه‌ی کوتاهترین مسیرها بین K رأس با خودشان را به دست بیاوریم که کار ساده‌تر باشد).

8.6.3 علی و امیر می‌خواهند از قم به مشهد بروند و چون مسیر طولانی است، می‌خواهند نوبتی رانندگی کنند و در هر استراحتگاه میان‌راهی جای خود را عوض کنند. چون علی حس جهت‌یابی بهتری دارد، باید در شهرهای مبدأ و مقصد او رانندگی کند و امیر فقط در جاده رانندگی می‌کند. نقشه‌ی مسیر به شکل یک گراف وزن‌دار جهت‌دار داده شده است که وزن یالهای آن مثبت است و رأسهای آن استراحتگاه‌ها و یالهای آن جاده‌های بین آنها هستند. الگوریتمی ارائه دهید که در صورت امکان، کوتاهترین مسیر بین قم و مشهد را پیدا کند، طوری که علی و امیر بتوانند نوبتی رانندگی کنند.

راه حل:

کوتاهترین مسیرها بین همه‌ی زوج رأسها را حساب می‌کنیم و تعداد یالهای آنها را هم علاوه بر طول آنها نگه می‌داریم. کوتاهترین مسیر با طول فرد بین قم و مشهد را در صورت وجود برمی‌گردانیم و اگر کوتاهترین مسیرها همه طول زوج داشتند جواب خیر (غیرممکن) است.

8.6.4 در یک شهر قوانین رانندگی عجیبی وجود دارد. این شهر تعدادی تقاطع دارد که بین آنها راه‌های دوطرفه بین تقاطع‌های متمایز است. چراغ‌های راهنمایی در هر تقاطع وجود دارند که رنگ آنها به صورت تناوبی مدتی آبی و مدتی قرمز است. در صورتی می‌شود در یک راه تردد کرد که وقتی به آخر آن می‌رسیم چراغ راهنمایی همرنگ وقتی باشد که در ابتدای آن هستیم (اگر چراغ همان لحظه عوض شود رنگ جدید آن ملاک است). در نقشه‌ی شهر زمان طی کردن همه‌ی جاده‌ها، زمان هر رنگ

هر چراغ راهنمایی و رنگ اولیه هر چراغ راهنمایی و زمان باقیمانده تا تغییر رنگ آن داده شده‌اند. (همه‌ی این اعداد صحیح هستند). شما باید مسیری را پیدا کنید که کمترین زمان برای رسیدن از یک مبدأ داده شده به یک مقصد داده شده را بدهد. اگر بیش از یک مسیر با چنین خصوصیاتی وجود دارد، یکی از آنها را پیدا کنید. در ضمن ماشین‌ها می‌توانند در یک تقاطع توقف کنند.

راه حل:

الگوریتم کوتاهترین مسیر بین همه‌ی رأسها را تغییر بدهید تا توقف در تقاطع‌ها را هم حساب کند. درایه متناظر مبدأ و مقصد جواب مسأله است.

8.6.5 گرافی وزندار و جهت دار $G = (V, E)$ بدون دور منفی را در نظر بگیرید . فرض کنید m حداکثر تعداد جفت رؤوس $u,v\in V$ از حداقل تعداد یال ها در کوتاه ترین مسیر از u به v باشد . (در اینجا کوتاه ترین مسیر وزندار مدنظر است نه تعداد یالها) در الگوریتم Bellman-Ford تغییری ایجاد کنید تا در m+1 بار عبور پایان یابد

راه حل:

ویژگی همگرایی نشان میدهد که برای هر راس v , الگوریتم کوتاه ترین مسیر d[v] را تخمین میزند مقدار نهایی انرا (هر کوتاهترین مسیر به v)بعد از تکرار الگوریتم BELLMAN-FORD بدست می اورد . بدین گونه پس از m بار تکرار , BELLMAN-FORD به اتمام میرسد . در ابتدا ما m را نمیدانیم , پس نمیتوانیم تکرار حلقه را دقیقا m بار انجام و سپس به ان خاتمه دهیم . ولی ما دقیقا وقتی هیچ چیز دیگر تغییر نمیکند الگوریتم را خاتمه میدهیم , و پس از m+1 تکرار پایان میپذیرد (پس از یک تکرار که هیچ تغییری اتفاقی نیافتاده)

```
BELLMAN-FORD-(M+1)(G,w,s)
INITIALIZE-SINGLE-SOURCE(G,s)
changes ← TRUE
while changes= TRUE
    do changes ← FALSE for each edge (u,v)∈ E[G]
        do RELAX-M(u,v,w)
RELAX-M(u,v,w)
if d[v] > d[u]+w(u,v)
    then d[v]←d[u]+w(u,v)
        π[v]←u
        changes ← TRUE
```

چک کردن عدم وجود حلقه منفی (بر این اساس که یک d وجود دارد که اگر یک relaxation دیگر انجام شود تغییر میکند)حذف شده به این علت که این الگوریتم از حلقه خارج نمیشود مگر انکه همه d ها تغییر نکنند.

8.6.6 گرافی جهت دار $G = (V, E)$ را بگیرید که در ان هریال $(u,v)\in E$ ارزشی برابر با $r(u,v)$ که عددی حقیقی بین 0 تا 1 است که قابلیت اطمینان ارتباط کانال از راس u به v را نشان میدهد دارد.ما $r(u,v)$ را بعنوان احتمال اینکه کانال از u به v شکست نخورد در نظر میگیریم و فرض میکنیم که این احتمالات مستقل هستند.الگوریتمی ارایه دهید که قابل اطمینان ترین مسیر بین دو راس را پیدا کند .

راه حل:

برای پیدا کردن قابل اطمینان ترین راه بین s و t الگوریتم دایکسترا را با وزن یالهای $w(u,v)=-\lg r(u,v)$ برای پیدا کردن کوتاه ترین مسیر از s در زمان $O(E+V \lg V)$ اجرا میکنیم . قابل اطمینان ترین مسیر کوتاه ترین مسیر از s به t و ارزش اطمینان ان برابر با حاصلضرب ارزش اطمینان یال های ان است .

به دلیل اینمه احتمالات مستقل هستند , احتمال این که یک مسیر شکست نخورد برابر است با حاصلضرب احتمالت اینکه یالهای ان شکست نخورند. ما میخواهیم مسیری از $s \xrightarrow{p} t$ به طوری که $\prod_{(u,v)\in p} r(u,v)$ حداکثر شود . این هم ارز است با اینکه عبارت $\lg(\prod_{(u,v)\in p} r(u,v)) = \sum_{(u,v)\in p} \lg r(u,v)$ را ماکزیمم کنیم که هم ارز است با مینیمم کردن عبارت $\sum_{(u,v)\in p} -\lg r(u,v)$. r(u,v)) میتواند صفر باشد و $\lg 0$ تعریف نشده است . پس در این الگوریتم $\lg 0 = -\infty$ تعریف میکنیم). همچنین اگر اوزان را برابر با $w(u,v)=-\lg r(u,v)$ قلمداد کنیم , مساله تبدیل به مساله کوتاهترین مسیر شده است .

چون $\lg 1 = 0, \lg x < 0$ for $0 < x < 1$ و $\lg 0 = -\infty$ را تعریف کردیم , همه ی اوزان w نامنفی هستند و حالا میتوانیم از دایکسترا برای پیدا کردن کوتاه ترین مسیر ها از s در زمان $O(E + V \lg V)$ استفاده کنیم .

روش دوم :

هم چنین میتوان از اصل احتمالات با اجرای ورژنی تغییر داده شده از الگوریتم دایکسترا که حاصلضرب قابلیت اطمینان ها در طول یک مسیر را ماکسیمم میکند بجای اینکه مجموع اوزان یال ها در طول یک مسیر را مینیمم کنیم

در دایکسترا ار قابلیت اظمینان بعنوان وزن یال ها استفاده کنید

- max (and EXTRACT-MAX) for min (and EXTRACT-MIN) in relaxation and the queue,
- × for + in relaxation,
- 1 (identity for ×) for 0 (identity for+) and −∞(identity for min) for∞(identity for max).

برای مثال الگوریتم پایین برای استفاده بجای عمل relax استفاده میشود

```
RELAX-RELIABILITY(u,v,r)
 if d[v] < d[u] ·r(u,v)
      then d[v]←d[u] ·r(u,v)
      π[v]←u
```

8.6.7 فرض کنید $G = (V, E)$ گرافی وزندار و جهت دار دارای تابع وزن $w : E \to R$ و راس مبدا s است . ثابت کنید برای همه یال ها $(u,v)\in E$ داریم : $\delta(s,v)\le \delta(s,u)+w(u,v)$.

راه حل:

فرض کنید که کوتاه ترین مسیر p مبدا s به راس v وجود دارد . پس p از هر مسیر دیگری از s به v کم وزن تر است . بویژه مسیر p از مسیر منحصر بفردی که کوتاه ترین مسیر از مبدا s به راس v را در اختیار دارد و سپس یال (u,v) را اختیار میکند وزن بیشتری ندارد .

8.6.8 گراف وزندار و جهت دار $G = (V, E)$ با راس مبدا s را فرض کنید و G تابع INITIALIZE-SINGLE-SOURCE(G,s) بروی ان فراخوانی شده است . ثابت کنید اگر دنباله ای از ریلکسیشن ها مقدار $\pi[s]$ را به non-NIL تغییر دهد , سپس G دارای دوری منفی است.

راه حل:

هرگاهRELAX , پدر (π) را برای راسی تعیین میکند همچنین مقدار d را کاهش میدهد . سپس اگر $\pi[s]$ به مقدار non-NIL ست شده d[s] از مقدار اولیه اش که صفر بوده به مقداری منفی میرسد . اما d[s] وزن مسیری از s به s باشد , که در واقع دوری شامل s است . پس دوری منفی داریم .

8.6.9 گراف وزندار و جهت دار $G = (V, E)$ که هیچ دور منفی ندارد را فرض کنید . راس $s \in V$ را راس مبدا در نظر بگیرید و روی G تابع INITIALIZESINGLE-SOURCE(G,s) اجرا شود . ثابت کنید که دنباله ای با $|V| - 1$ بار ریلکسیشن که $d[v] = \delta(s,v)$ را برای همه ی $v \in V$ تولید میکند .

راه حل:

فرض کنید درخت کوتاه ترین مسیر هارا که اسمش $G(\pi)$ است را در اختیار داریم . یال های داخل $G(\pi)$ را بر اساس ترتیبی که BFS اونهارو ملاقات میکنه ریلکس میکنیم . سپس ما تضمین میکنیم که یال های هر کوتاه ترین مسیر به ترتیب ریلکس شده اند.بر اساس ویژگی path-relaxation برای هر $v \in V$ داریم $d[v] = \delta(s,v)$. چون $G(\pi)$ حداکثر $|V| - 1$ یال را شامل میشود , ما فقط نیاز داریم تا $|V| - 1$ یال را ریلکس کنیم تا برای همه راس ها $d[v] = \delta(s,v)$ بشود

8.6.10 گراف وزندار و جهت دار را فرض کنید . همچنین کوتاه ترین مسیر از راس s به راس t داده شده . اگر وزن هر یال را 10 واحد زیاد کنیم ایا کوتاه ترین مسیر در گراف جدید همان مسیر قبلی میماند ؟

راه حل:

کوتاه ترین مسیر ممکن است تغییر کند . به این دلیل که امکان دارد که در مسیر های گوناگون از s به t تعداد یال ها متفاوت باشد . برای مثال، کوتاه ترین مسیر دارای وزن 15 می باشد دارای 5 یال . فرض کنید مسیر دیگری با 2 یال و وزن کل 25باشد . وزن کوتاه ترین مسیر $5 * 10$ افزایش یافته است و می شود $15 + 50$. وزن مسیر دیگر $2 * 10$ افزایش یافته و $25 + 20$. بنابراین کوتاه ترین مسیر به مسیر دیگر با وزن 45 تغییر می یابد.

8.6.11 این شبیه به سوال فوق است. آیا کوتاه ترین مسیر با ضرب وزن همه یال ها در 10 تغییر می یابد ؟

راه حل:

اگر ما وزن تمام یال ها را ضربدر 10 کنیم ، کوتاه ترین مسیر تغییر نمی کند. دلیل آن ساده است ، وزن تمام مسیرهای از s به t توسط همان مقدار ضرب شده است. تعداد یال ها در یک مسیر مهم نیست. این مانند تغییر واحد وزن است.

8.6.12 گرافی جهت دار داده شده که در آن هر یال دارای وزن 1 یا 2 است ، کوتاه ترین مسیر از یک راس مبدا s را به یک راس مقصد t پیدا کنید . انتظار می رود پیچیدگی زمانی $O(V + E)$ شود.

راه حل:

اگر ما الگوریتم کوتاه ترین مسیر دیکسترا را اجرا کنیم ، می توانیم کوتاه ترین مسیر را در زمان $O(E + V\text{Log}V)$ پیدا کنیم . چگونه آن را در زمان $O(V + E)$ پیدا کنیم؟ ایده این است که از BFS استفاده کنیم . یکی از مشاهدات مهم در مورد BFS است، راه مورد استفاده در BFS همیشه حداقل تعداد یالهای بین هر دو راس را دارد . بنابراین اگر همه یال ها وزنشان یکی باشد ، ما می توانیم از BFS برای پیدا کردن کوتاه ترین مسیر استفاده کنیم. برای این مشکل، ما می توانیم نمودار تغییر و تقسیم تمام لبه وزن 2 به دو لبه وزن 1 هر یک از. در نمودار اصلاح شده ، ما می توانیم برای پیدا کردن کوتاه ترین مسیر گراف را بهینه کنیم به این صورتکه یالهای با وزن دو را به دو یال با وزن 1 تبدیل کنیم و در گراف بهینه میتوانیم از BFS برای پیدا کردن کوتاه

ترین مسیر استفاده کنیم. این رویکرد چگونه $O(V + E)$ است؟ در بدترین حالت، تمام یال ها دارای وزن 2 هستند و ما نیاز به انجام $O(E)$ عملیات برای تقسیم تمام یال ها داریم ، به طوری که پیچیدگی زمانی می شود: $O(E) + O(V + E)$ که برابر است با : $O(V + E)$

8.6.13 در یک بازار ارز برای تبدیل هر دو واحد پول یک ضریب وجود دارد . مثلا اگر ضریب تبدیل واحد پول a به واحد پول b برابر 3 باشد میتوان 10 واحد از a را به 30 واحد از b تبدیلکرد.دقت دقت کنید که نرخ تبدیل میتواند غیر صحیح باشد (مثلا 1/5)الگوریتمی ارایه دهید تا یک چرخه سود اور را در صورت وجود پیدا کند (چرخه ای که بتوان با شروع ار مبلغی پول به واحد a و با تعدادی تبدیل نرخ به مقدار بیشتری پول در واحد a دست یافت)

راه حل:

مساله را با گراف مدل سازی می کنیم به این شکل که رئوس را برابر واحدها قرار میدهیم و وزن یالها را به این شکل قرار میدهیم که اگر نرخ یک تبدیل برابر w باشد وزن یال متناظر را $1/w$ قرار میدهیم یک چرخه نیز معادل یک دور در گراف خواهد بود.چرخه ی سودآوری چرخه ایست که حاصل ضرب یال های آن کوچک تراز ۱باشد. معنا این جمله این است که حاصل جمع لگاریتم وزن یال ها منفی باشد.حال از الگوریتم Bellman Ford برای پیداکردن دور منفی استفاده می کنیم به این صورت که اگر در آخرین دفعه ی استفاده از Bellman Ford برای فهمیدن دور منفی فهمیدیم که v جزو این دور است . از v شروع میکنیم و به $\pi(v)$ میرویم و همین منوال را ادامه میدهیم تا دوباره به v برسیم . به این شکل دور منفی را پیدا خواهیم کرد .

8.6.14 الگوریتمی بهینه ارائه دهید تا در یک گراف جهت دار با وزن مثبت ، کوتاه ترین مسیر از s به t را حساب کند با در نظر گرفتن اینکه اجازه دارید وزن یکی از یال ها را صفر کنید . هزینه ی زمانی الگوریتم ارائه شده را نیز محاسبه کنید .

راه حل:

ابتدا با اجرای الگوریتم دایکسترا بر روی گراف ، کوتاه ترین مسیر از s به تمام رؤوس را محاسبه میکنیم . سپس با اجرای دوباره الگوریتم دایکسترا کوتاه ترین مسیر از تمام رؤوس تا t را محاسبه میکنیم . سپس به ازای هر یال (u,v) مقدار $d(s,u)+d(v,t)$ را حساب میکنیم و بین این مقادیر مینیمم میگیریم . هزینه زمانی الگوریتم هم $O(E + V)\log v$ خواهد بود .

8.6.15 بدون تغییرالگوریتم دایکسترا،الگوریتمی ارائه دهید تا در یک گراف که وزن یال ها عددی صحیح میباشد و در صورت وجود چندین کوتاه ترین مسیر از s به vمسیر با تعداد یال کمتر را پیدا کنیم درستی الگوریتم خود ر اثبات کنید .

راه حل:

ایده ی اصلی این است که به تمام یال ها وزن مساوی اضافه کنیم و در این بین دو مسیر با طول برابر ، مسیری که دارای تعداد یال بیشتری است ، درنهایت طول بیشتری پیدا خواهد کرد.اما مشکلی که وجود دارد این است کهممکن است یک کوتاه ترین مسیر بعد از این تغییرات دیگر کوتاه ترین مسیر نباشد . نکته این است که باید به وزن هر یال حداکثر $1/(v\text{-}1+\varepsilon)$ واحد اضافه کنیم چرا که طول یک مسیر حداکثر v-1 است و اینگونه تضمین میکنیم که طول هر مسیر کمتر از 1 واحد افزایش پیدا میکند . با توجه به اینکه وزن یالها عددی صحیح می باشد تفاوت طول دو مسیر حداکثر 1 میباشد . لذا امکان ندارد مسیر دلخواه s1 که طولش کمتر از مسیر دلخواه s2 بود بعد از اعمال این روش طولش از s2 بیشتر شود و همچنان کوتاه ترین مسیر باقی می ماند . بعد از اعمال این تغییر ، الگوریتم دایکسترا را روی گراف تغییر یافته اعمال می کنیم.

8.6.16 گراف G داده شده است . تعداد کوتاه ترین مسیر های موجود در این گراف را بیابید .

راه حل:

الگوریتم فلوید-وارشال را اینگونه تغییر میدهیم: در هر گام الگوریتم علاوه بر طول کوتاهترین مسیر پیدا شده تعداد مسیرهای با این طول را نیز نگه میداریم (در ماتریس count) برای گام بهروزرسانی اینگونه عمل میکنیم: اگر طول مسیری از که از a به

b از طریق v میرود کوتاه تر از مقدار فعلی بود , $count[a][b] = count[a][v] \times count[v][b]$ قرار میدهیم، اگر طول مسیر جدید مساوی مقدار قبلی بود ، $count[a][b] = count[a][b]+count[a][v]\times count[v][b$ قرار میدهیم .

8.6.17 گراف G داده شده است . با استفاده از الگوریتم فلوید-وارشال وجود دور منفی در G را بررسی کنید .

راه حل:

ادعا میکنیم گراف دور منفی دارد اگروتنها اگر پس از اجرا الگوریتم فلوید-وارشال اگر مقدار distance[v][v] به ازای حداقل یک راس منفی باشد.یک سمت ادعا واضح است، درصورتی که چنین راسی موجود باشد گراف دور منفی دارد.ازسمت دیگر ، یک دور منفی در گراف را در نظر میگیریم . دنباله رووس موجود دراین دور را v(k) ... v(1) مینامیم به گونه ای که v(k) راس با بیشترین شماره است.هنگامی که در زمان اجرای الگوریتم فلوید-وارشال راس میانی برابر v(k) است . مقدار distance[v(1)][v(k-1)] حداکثر برابر با طول مسیر v(k) ... v(1) است . سپس با بروزرسانی از طریق v(k) مقدار distance[v][v] با توجه به منفی بودن طول دور یک عدد منفی میشود .

8.6.18 گراف G داده شده است طول کوتاه ترین دوری در G که تعدادی فردی یال دارد را بیابید .

راه حل:

ابتدا طول کوتاه ترین مسیرهای با زوج یال و فرد یال را بین هرجفت راس پیدا میکنیم. برای این کار الگوریتم فلوید-وارشال را اینگونه تغییر میدهیم: بجای نگه داشتن یک ماتریس شامل طول کوتاه ترین مسیرهای بین رووس تا مرحله فعلی ، یک ماتریس شامل طول کوتاه ترین مسیر با زوج یال و یک ماتریس شامل طول کوتاهترین مسیر با فرد یال نگه میداریم ،برای اشاره به این دو ماتریس به ترتیب sop[u][v] و sep[u][v] را استفاده میکنیم.مقادیر اولیه این دو ماتریس به شکل های زیرند:

$$sop[u][v] = w[u][v] \ , \text{if } u \neq v \ , sop[u][u] = Inf$$

$$sep[u][v] = inf \ , \text{if } u \neq v \ , sep[u][u] = ۰$$

برای گام بروزرسانی اینگونه عمل میکنیم:

$$sop[a][b] = \min(sop[a][b], sep[a][v] + sop[a][v], sop[a][v] + sep[a][v])$$

$$sep[a][b] = \min(sep[a][b], sep[a][v] + sep[v][b], sop[a][v] + sop[v][b])$$

سپس طول کوتاه ترین دور فرد به شکل زیر پیدا میشود :

$$\min_{u,v,w \in v(G)} (sop[u][v]+sop[v][w]+sop[w][u], sep[u][v]+sep[v][w]+sop[w][u])$$