"بسم الله الرحمن الرحیم"

سیدمانی پرشعبانی ۸۱۰۸۰۱۰۸۰
معین اکبری شهیر ۸۱۰۸۰۱۰۵۵
مهسا کرمیان ۸۱۰۸۰۱۰۷۴

دانشگاه تهران

دانشکدگان فنی

دانشکده علوم مهندسی

طراحی الگوریتم، پاییز ۱۴۰۳

تمرین‌های الگوریتم‌های گراف

۱. "علوم‌مهندسی آباد" شهریست شامل $n \leq 10^5$ میدان و m خیابان مستقیم میان بعضی از جفت-میدان‌ها (گراف ساده). بابک که به تازگی شهردار شده است قصد دارد تعدادی از میدان‌های شهر را گل‌کاری کند به طوری که هر میدان یا گل‌کاری شده باشد یا به میدانی که گل‌کاری شده است، خیابان داشته باشد. بنابر پاره‌ای از مشکلات، او حداکثر می‌تواند $\left\lfloor \frac{n}{2} \right\rfloor$ تا از میدان‌ها را انتخاب کند. به او در این امر کمک کنید و الگوریتمی ارائه دهید که میدان‌های انتخابی را خروجی دهد. همچنین آن را از لحاظ زمانی تحلیل کنید.

۲. یک گراف n راسی و m یالی داریم. روی هر یال آن یک عدد اعشاری وجود دارد. الگوریتمی ارائه دهید که مشخص کند آیا در این گراف دوری وجود دارد که حاصل ضرب تمامی یال‌هایش از 1 بزرگتر باشد؟ $(nm \leq 10^6)$

۳. ماتریس D کوتاه‌ترین مسیر بین هر دو راس در گراف وزن‌دار G را نگهداری می‌کند به طوری که $D[u, v]$ کوتاه‌ترین مسیر از راس u به راس v است. فرض کنید وزن یک یال در گراف G از w_e به $\acute{w}_e$ تغییر پیدا کند. الگوریتمی از مرتبه زمانی $O(n^2)$ طراحی کنید که مقادیر ماتریس D را به‌روزرسانی کند.

۴. یک گراف وزن دار که در آن وزن هر یال ۱ یا ۲ است را در نظر بگیرید، کوتاه ترین مسیر را از یک راس منبع داده شده s تا راس مقصد t پیدا کنید به طوری که پیچیدگی زمانی آن از $O(V + E)$ باشد.

۵. فرض کنید T یک زیر درخت فراگیر کمینه از G و $\acute{T}$ زیردرخت فراگیر دیگر از G باشد هر حرکت یک یال از $\acute{T}$ را با یک یال از T جایگزین میکند. الگوریتمی ارائه دهید که با دنباله ای از حرکات، $\acute{T}$ را به T تبدیل کند با این شرط که با هر تغییری که انجام میدهیم همچنان درخت، فراگیر باشد و مجموع وزن یالهای آن هیچوقت بیشتر نشود.

تمرین‌های الگوریتم‌های بیشینه جریان

۶. فرض کنید M یک جدول x * y باشد که در هر خانه ای جدول یک عدد حقیقی نامنفی وجود دارد. به طوری که مجموع اعداد هر سطر و ستون عددی صحیح است. ثابت کنید جدولی x * y وجود دارد به طوری که در هر خانه از جدول یک عدد صحیح نامنفی وجود دارد و مجموع اعداد هر سطر و ستون آن همانند جدول M است.

۷. یک گراف جهت‌دار $G = (V, E)$ (تصور کنید یک شبکه از جاده‌ها) به شما داده شده است. مجموعه‌ای از گره‌ها $X \subset V$ به عنوان گره‌های پر جمعیت و مجموعه دیگری $S \subset V$ به عنوان گره‌های *امن* مشخص شده‌اند (فرض کنید که X و S اشتراکی ندارند). در صورت وقوع شرایط اضطراری، می‌خواهیم مسیرهای تخلیه‌ای از گره‌های پر جمعیت به گره‌های امن طراحی کنیم.

یک مجموعه از مسیرهای تخلیه به صورت مجموعه‌ای از مسیرها در G تعریف می‌شود به گونه‌ای که:

- هر گره در X، نقطه شروع یک مسیر باشد.
- آخرین گره در هر مسیر در S قرار داشته باشد.

- مسیرها هیچ یال مشترکی نداشته باشند.

چنین مجموعه‌ای از مسیرها به ساکنان گره‌های پر جمعیت اجازه می‌دهد که به S فرار کنند، بدون اینکه باعث ازدحام بیش از حد در هیچ یال G شوند.

الف) با داشتن G، X، و S، نشان دهید که چگونه می‌توان در زمان چندجمله‌ای تصمیم گرفت که آیا چنین مجموعه‌ای از مسیرهای تخلیه وجود دارد یا خیر.

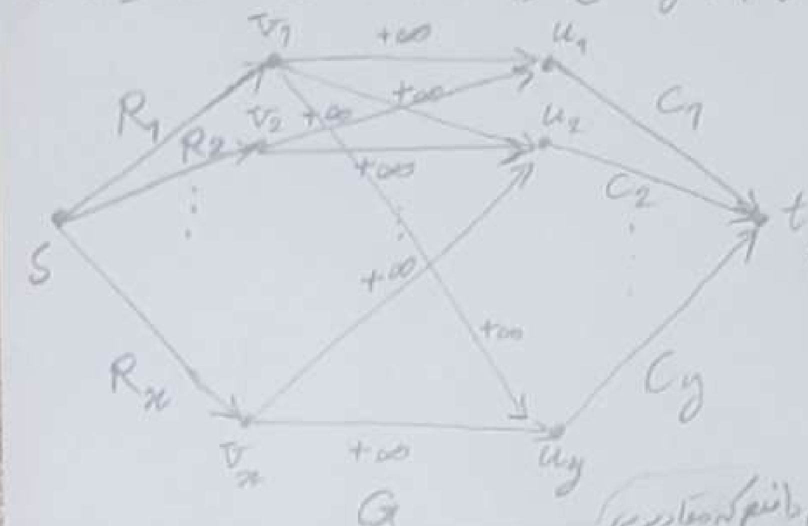
ب) فرض کنید دقیقاً همان مسئله‌ای که در بخش (الف) بیان شد را داریم، اما می‌خواهیم شرط سوم را قوی‌تر کنیم. بنابراین، شرط سوم تغییر می‌کند و می‌گوید: «مسیرها هیچ گره مشترکی نداشته باشند».
با این شرط جدید، نشان دهید که چگونه می‌توان در زمان چندجمله‌ای تصمیم گرفت که آیا چنین مجموعه‌ای از مسیرهای تخلیه وجود دارد یا خیر.

ج) یک مثال ارائه دهید که در آن برای یک گراف G، مجموعه X و S، پاسخ بخش (الف) *بله* باشد اما پاسخ بخش (ب) خیر باشد.

۸. یک شرکت تولید نرم افزار می‌خواهد برای به‌کارگیری نیروهای خود در پروژه‌های سال آینده اش برنامه‌ریزی کند. این شرکت در سال آینده n پروژه در دست دارد که به ترتیب انجام خواهند شد. برای هر پروژه تعدادی نقش تعریف می‌شود. مثلا برای پروژه i ام k_i نقش وجود دارد. این برنامه‌ریزی باید به گونه ای باشد که برای هر نقش در هر پروژه یک نفر مشغول به کار باشد. ممکن است افراد به هر دلیلی نتوانند در نقشی خاص از پروژه‌ای خاص کار کنند. با استفاده از شبکه جریان الگوریتمی طراحی کنید که این برنامه ریزی را با در نظر گرفتن محدودیت‌های زیر انجام دهد یا تشخیص دهد امکان برنامه ریزی وجود ندارد.

- هر یک از نیروها حداکثر در c پروژه حضور داشته باشند.
- در هر پروژه هر نفر حداکثر یک نقش داشته باشد.

جمع سطر i ام را R_i و جمع ستون j ام را C_j می نامیم و شبکه ای با ظرفیت های زیر می طرحیم:



G

تمامی v_i ها $x \geq i \geq 1$ به u_j ها با یال هایی با ظرفیت نامتناهی متصل هستند. ($1 \leq j \leq y$)

لم 1: اگر حداکثر جریان f،
$$f = \begin{cases} R_i & (s, v_i) \\ C_j & (u_j, t) \end{cases}$$
باشد آنگاه جدول $y \times x$ با مقادیر صحیح و مجموع سطر و ستون یکسان با جدول قبلی وجود دارد.

$f(v_1,u_1)$	$f(v_1,u_2)$	$\cdots$	$f(v_1,u_y)$
$f(v_2,u_1)$	$\cdots$	$\cdots$	$\cdots$
$\vdots$			
$f(v_x,u_1)$	$\cdots$	$\cdots$	$f(v_x,u_y)$

اثبات: فرض می کنیم جریانی از s به t وجود دارد؛ می دانیم که مقادیری که این تابع جریان به یال ها نسبت می دهد صحیح هستند. از روی این جریان f جدول را به این صورت می سازیم: اگر A_{ij} نشان دهنده خانه سطر i ام و ستون j ام جدول باشد، داریم (f تابع جریان است)

$$A_{ij} = f(v_i, u_j)$$

جمع ستون j: $\sum_{i=1}^{x} A_{ij} = \sum_{i=1}^{x} f(v_i, u_j)$: جریان ورودی به راس u_j = جریان خروجی از راس u_j = C_j

جمع سطر i: $\sum_{j=1}^{y} A_{ij} = \sum_{j=1}^{y} f(v_i, u_j)$: جریان خروجی از راس v_i = جریان ورودی به راس v_i = R_i ∎

لم 2: می توان حداکثر جریانی معادل
$$f = \begin{cases} R_i & (s, v_i) \\ C_j & (u_j, t) \\ k_m & o.w. \end{cases}$$
از s به t عبور داد.

اثبات: می توان از جدول اولیه، به یک جریان معتبر (با مقادیر حقیقی ولی $|f|$ صحیح) رسید (با انجام معکوس مراحل لم قبل) پس از G حداقل جریان f قابل گذراندن است پس Max Flow جریانی با حداقل $|f|$ پیدا می کند و
$$f' = \begin{cases} R_i & (s, v_i) \\ C_i & (u_j, t) \\ k_m & \text{بسبب} \end{cases}$$
حالتی است که تمامی یال های (s, v_i) و (u_j, t) و $|f'| = |f|$ ∎

از لم 1 و 2 نتیجه می گیریم که جدولی با مقادیر صحیح و مطابق شرایط مسئله وجود دارد.

7.

الف) برای اینکه بتوان سوال را با روش Max flow حل کرد لازم است که Source

و Sink داشته باشیم به همین خاطر تمام رئوس x که در گراف G قرار دارند را به

Source "M" وصل می‌کنیم و به هر یال متصل شده از x به S ظرفیت ∞ می‌دهیم.

به همین ترتیب برای گره‌های S، از گره‌های S به گره Sink "T" متصل می‌کنیم و به

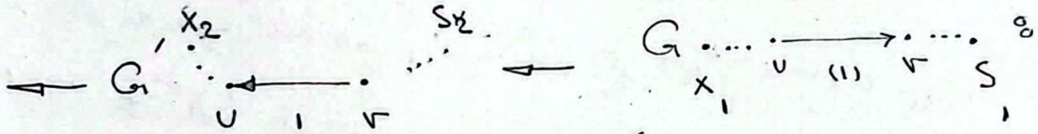
هر یال آن ظرفیت بی‌نهایت نسبت می‌دهیم. چرا که گفته شده: "از هر مسیر تنها یکبار بتوان

عبور کرد" را تضمین کنیم لازم است که برای تمام مسیرهای گراف ظرفیت را یک قرار دهیم که در صورت

هر بار عبور از آن با صفر شدن ظرفیت دیگر نتوان از آن عبور کرد.

(۷,۳)

اثبات



گراف باقیمانده را رسم می‌کنیم اگر از مسیر (۷,۳) گذشته باشیم در گراف $G_{f,p}$ باید

یال به ظرفیت ۱ از (۳,۷) رسم نماییم

حالت اول: در مسیر یافته شده p در گراف G' از یال (۳,۷) عبور نمی‌کنیم

حالت دوم: از مسیر (۳,۷) عبور می‌کنیم و چون یکبار در جهت (۷,۳) حرکت کرده و بار دیگر در جهت (۳,۷)

در صورت رسم گراف متناظر با G' گویا از آن یال اصلاً عبور نکرده‌ایم

⇐ در هر دو حالت شرط نقض نمی‌شود.

الف) سپس برای این گراف بیشینه‌یافته الگوریتم "Edmons karp" که در زمان $O(VE^2)$

تعداد رأس‌های ایکس

قابل اجراست اعمال می‌کنیم درصورتی که اندازه جریان ما $|f|$ مساوی با $\boxed{n(x)}$ تعداد

رأس‌های x باشد یعنی به ازای هر x یک مسیر به S وجود دارد که با مسیرهای دیگر تداخل

ندارد

ب) برای آنکه از هر کوه فقط یک بار عبور کنیم لازم است هر کوه را به یک یال به اندازه ۱ تبدیل کنیم در این صورت با استدلال صفحه قبل تنها کیلیا میتوان از این کوه عبور کرد.

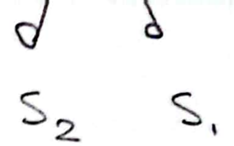
در نتیجه برای حل این سوال شروط ایجاد یک منبع Source و Sink + ظرفیت تمامی یال های گراف به اندازه ی ۱ را در کنار شرط با x قرار می دهیم و دوباره الگوریتم Edmons karp

$E \leq V^2$

را با $O(VE^2)$ محاسبه می کنیم که بر اساس چند جمله ای است در صورتی که اندازه ی جریان برابر با $n(x)$ شود یعنی می توان به ازای هر راس یک مسیر همتا به S یافت.

علت آنکه اندازه ی جریان باید برابر با $n(x)$ شود تا الگوریتم الگوریتم درست کار می کند آن است که به ازای هر شهر x یک تخلیه باید به یک شهر کد انجام دهیم! می دانیم که هر مسیر ظرفیت تمامی یال های آن ۱ است پس در صورت کارکرد درست الگوریتم به ازای هر x یک جریان با ظرفیت ۱ به کد میل می کند و خروجی تمامی کوه ها کد ها ۱ است و چون از هر کوه کد به کوه tank یک یال وصل می شود پس اندازه ی جریان برابر است با xn

الگوریتم "Edmons karp": شیوه حل این الگوریتم به این صورت است که از راس کد به راس t یک مسیر با کمک "BFS" می یابیم، مسیری که به BFS بدل می شود همواره به روی کمترین تعداد یال تحرک می کند و دچار مشکل درست و افزایش پیچیدگی الگوریتم که ممکن بود در روش DFS به آن برخورد کنیم نخواهیم شد. پس با یافتن مسیر و انتخاب کمترین ظرفیت یال موجود در مسیر، گراف باقیمانده را رسم می کنیم و دوباره سعی می کنیم با کمک BFS یک مسیر از کد به t بیابیم، در نهایت این پروسه را آنقدر ادامه می دهیم که دیگر نتوانیم مسیری در گراف باقیمانده بیابیم

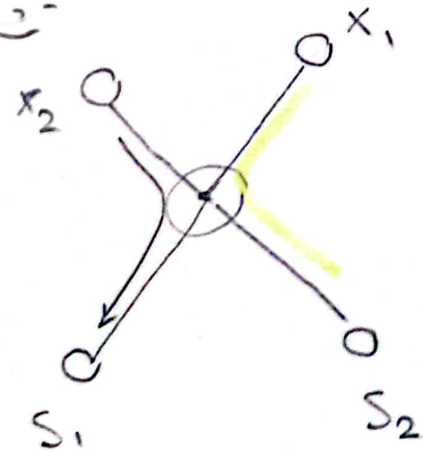


برای نوشتن گراف باقی‌مانده

$$C_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \\ f(v,u) & \text{if } (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

ج) آیا مسیری وجود دارد که در هیچ یالی مشترک نباشد اما در رأسی مشترک باشد؟

بله

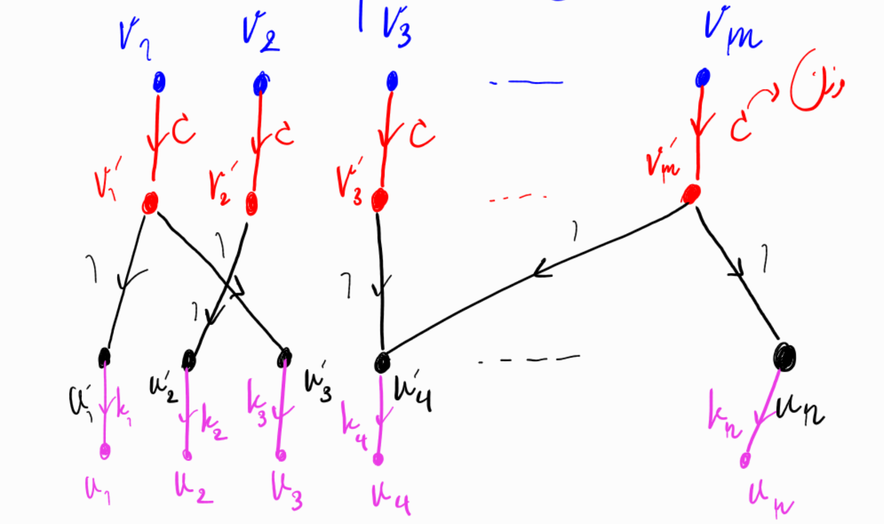


۸. یک شرکت تولید نرم افزار می‌خواهد برای به‌کارگیری نیروهای خود در پروژه‌های سال آینده اش برنامه‌ریزی کند. این شرکت در سال آینده n پروژه در دست دارد که به ترتیب انجام خواهند شد. برای هر پروژه تعدادی نقش تعریف می‌شود. مثلا برای پروژه i ام k_i نقش وجود دارد. این برنامه‌ریزی باید به گونه ای باشد که برای هر نقش در هر پروژه یک نفر مشغول به کار باشد. ممکن است افراد به هر دلیلی نتوانند در نقشی خاص از پروژه‌ای خاص کار کنند. با استفاده از شبکه جریان الگوریتمی طراحی کنید که این برنامه ریزی را با در نظر گرفتن محدودیت‌های زیر انجام دهد یا تشخیص دهد امکان برنامه ریزی وجود ندارد.

- هر یک از نیروها حداکثر در c پروژه حضور داشته باشند.
- در هر پروژه هر نفر حداکثر یک نقش داشته باشد.

مسئله را به این صورت تبدیل به یک مسئله max flow می‌کنیم:

به ازای هر فرد یک رأس می‌گذاریم که یک یال با وزن c از آن به یک رأس کمکی رفته است. این یال کمک می‌کند مطمئن شویم هر نفر حداکثر در c پروژه‌هاست. سپس به ازای هر پروژه هم یک رأس کمکی دیگر رأس می‌گذاریم. از رأس کمکی هر فرد تمام پروژه‌هایی که آن فرد می‌تواند در آن‌ها نقش داشته باشد یال با وزن 1 می‌گذاریم. (به رئوس کمکی پروژه‌ها (مثلا



در گراف تشکیل شده عددی که روی یال‌های قرمز در جواب max flow می‌آیند نشان دهنده این هستند که فرد i ام در چند پروژه کار خواهد کرد (که حداکثر c تا است)

و یال‌های سیاه نشان می‌دهند که یک فرد در کدام پروژه‌های می‌تواند کار کند (مثلاً (v_i', u_j') نشان می‌دهد که فرد i ام در پروژه j ام می‌تواند کار کند)

به طور خلاصه اگر یک فرد (i) در کار j کار کند ما یک جریان از v_i به u_j می‌گذاریم و برای اینکه هر کسی در حداکثر در c پروژه کار کند رئوس کمکی قرمز را اضافه کردیم! و برای اینکه مطمئن شویم هر نفر در هر پروژه حداکثر 1 کار دارد یال‌های سیاه رنگ را با وزن 1 می‌گذاریم!

به طور مشابه هم یال‌های صورتی رنگ تضمین می‌کنند که جریان ورودی به پروژه i ام (تعداد افرادی که در پروژه i کار می‌کنند) از k_i بیشتر نشود!

پس اگر یک Max flow در این گراف پیدا کنیم، معادل این است که به افراد کارها را Assign کنیم! و واضح است که اگر این جریان که بیشینه هم هست یال های صورتی را پر نکند، هیچ جریانی تقسیم کاری این کار را نمی کند!

* برای این که به روش Max-flow ادمونت کارپ و با چندین Source و Sink پیدا کنیم باید یک رأس به عنوان Source اصلی اضافه کنیم و به بقیه Source ها وصل کنیم و یک رأس هم به عنوان Sink اصلی داشته باشیم و از بقیه Sink ها به آن وصل کنیم!

اردر زمانی: به تعداد $n+m$ رأس داریم (تعداد کارمندان + تعداد پروژه ها) پس اردر اجرای Edmond Karp $O(VE^2)$ است که در اینجا می شود $O((n+m)(n+m+e))$

$e =$ تعداد یال های شکلی = ورودی در طبق افراد اینکه چه کار هایی می توانند بکنند