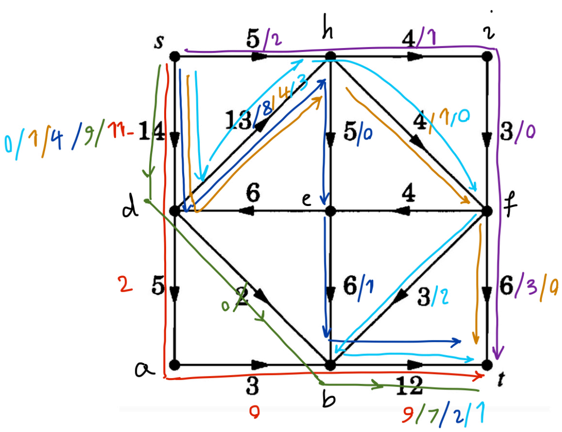
به نام خدا

# دانشگاه تهران، دانشکده مهندسی برق و کامپیوتر
# تحلیل و طراحی الگوریتم‌ها

راه حل تمرین کتبی پنجم (شبکه جریان)
طراح: علی کرامتی پور (alikeramatipour@gmail.com)

---

۱. در هر مرحله یک مسیر افزایشی مشخص می‌کنیم و ظرفیت باقی مانده یال‌ها را مشخص می‌کنیم. مسیر های افزایشی با رنگ‌های مختلف مشخص شده‌اند. همچنین ظرفیت هر مسیر نیز نوشته شده. مسیر افزایشی تا موقعی انتخاب شده که دیگر مسیری بین s و t موجود نباشد:




3 + 2 + 3 + 5 + 3 + 1 = 17

sdabt
sdbt
shift
sdhebt
sdhft
sdhfbt

mincut = { if , hf , he , db , ab } = 17

۲. در شبکه جریان $G = (V, E)$ به ازای هر راس $v \in V$، دو راس جدید $v_{in}$ و $v_{out}$ قرار می‌دهیم. این دو راس را با یال $(v_{in}, v_{out})$ بهم متصل می‌کنیم و ظرفیت یال را معادل ظرفیت راس $v$ می‌گذاریم. همچنین یال‌های به صورت $(u, v)$ را با $(u, v_{in})$ و یال‌های به صورت $(v, u)$ را با $(v_{out}, u)$ جایگزین می‌کنیم در آن که $u \in V$. در شبکه جریان جدید به جفت راس جدید $v_{in}$ و $v_{out}$، بیشتر از ظرفیت راس $v$ جریان وارد نمی‌شود و شرط خواسته شده برقرار می‌شود.

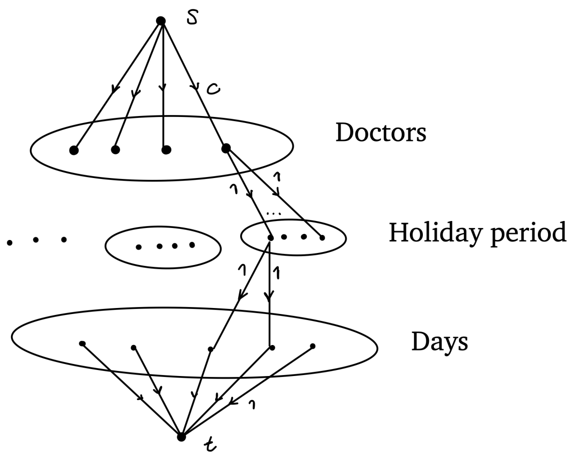
۳. مسئله را با گراف مدل می‌کنیم. مشهد و تهران را با رئوس s و t نام گذاری می‌کنیم.

(آ) ورودی مساله مشخصا یک شبکه جریان است. فرق آن این است که هر یال دوطرفه است که برای هر یال دو یال رفت و برگشت بین دو راس دو سر آن میگذاریم. باید یک مجموعه یال انتخاب کنیم تا هر مسیری از s به t شامل حداقل یک یال باشد. این مجموعه یک cut(s،t) است زیرا با حذف آن نباید مسیر دیگری از مبدا تا مقصد باشد و در دو مولفه جدا قرار بگیرند. همچنین هر cut(s،t) نیز حتما یک مجموعه یال معتبر برای جواب سوال است، زیرا اگر همه یال های کات را انتخاب کنیم هر مسیری حداقل از یکی از این یال ها باید استفاده کند. پس ثابت کردیم که این مجموعه متناظر یک کات های بین این دو راس هستند. پس حالا که مجموه با کمترین اندازه را می خواهیم هدف پیدا کردن min-cut است. همچنین طی قضیه ای که در کلاس ثابت کردیم می دانیم که max-flow=min-cut پس شار بیشینه بین s و t را محاسبه می کنیم و خروجی می دهیم.

(ب) پیچیدگی زمانی الگوریتم max-flow

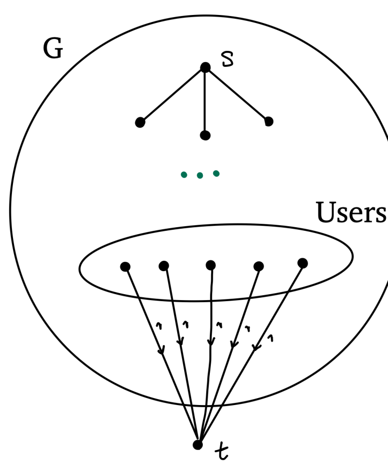
اگر از الگوریتم فوردفالکرسون استفاده کردید برابر O((n+m).F) است

۴. ابتدا شبکه جریان زیر را از روی داده‌های مسئله می‌سازیم:




یک راس به ازای هر پزشک در بخش اول قرار داده می‌شود. از راس منبع، به راس مربوط به هر پزشک، یالی با ظرفیت c قرار می‌دهیم. در بخش دوم گراف، به ازای هر پزشک، بخشی k راسی قرار داده می‌شود که هر راس نماینده یک بازه تعطیلات می‌باشد که به راس‌هایی از بخش سوم (روزها) متصل است، که پزشک توانایی حضور در آن روز را داشته باشد. در نهایت از هر روز یالی با ظرفیت یک به راس t (sink) متصل می‌کنیم. حال الگوریتم شار بیشینه را روی شبکه بالا اجرا کرده، در صورتی که شار بیشینه برابر تعداد روزهای تعطیل بود، یعنی برنامه‌ریزی موجود است که شرایط مسئله را برقرار می‌کند. زیرا: جریانی که از هر روز می‌گذرد، از یک و دقیقا یک پزشک گذشته. راس پزشک، در بیشتر از c جریان نمی‌تواند حضور داشته باشد. بنابراین c و یا کمتر از c روز کشیک خواهد بود. هر پزشک در هر بازه حداکثر یک روز کشیک می‌باشد(به دلیل یال با ظرفیت یک بین پزشک و بازه موردنظر). از آنجا که شروط مسئله برقرار می‌باشد، و شار بیشینه این شبکه برابر با بیشینه تعداد روز است، در صورت موجود بودن برنامه‌ریزی این شبکه آن را در اختیار ما می‌گذارد.

۵. شبکه G داده شده را به صورت زیر تغییر می‌دهیم:




راس s (سرور)، راس منبع در شبکه جریان می‌باشد. همچنین راس t را به شبکه اضافه کرده که راس مقصد می‌باشد. به ازای تمامی $u \in U$ یالی با ظرفیت یک به t متصل می‌کنیم. حال می‌خواهیم بیشینه وزن لینک‌ها (یال‌ها) را کمینه کنیم، تا شار بیشینه برابر U شود، یعنی جریانی از سرور به هر کاربر ایجاد بشود. از آنجا که بیشینه وزن مورد توجه پرسش است، می‌توانیم ظرفیت همه یال ها را برابر مقدار بیشینه در نظر بگیریم. این مقدار بیشینه از U، در صورتی که جوابی موجود باشد، بیشتر نخواهد بود (چرا؟). بنابراین با جستجو باینری در بازه [U،۱] می‌توانیم پاسخ را بیابیم. مرتبه زمانی از $O(Flog(\text{U}))$ خواهد بود.