به نام خدا

مرضیه علیداری - 810101236

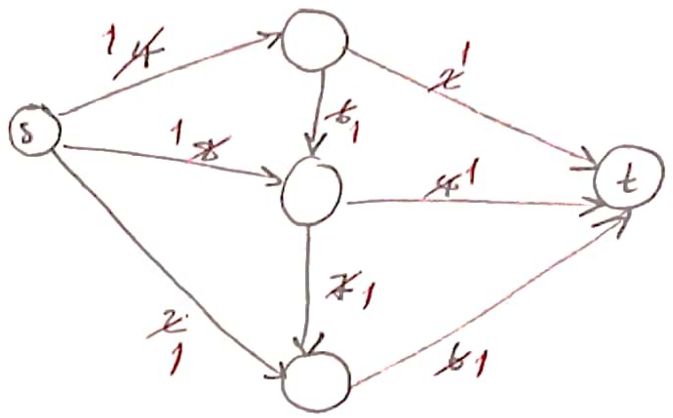
تکلیف چهارم الگوریتم پیشرفته

(1) پیدا کردن مسیرهای یال مجزا، یکی از کاربردهای الگوریتم شبکه شار maximum است.

ابتدا در شبکه ی شار، به هر کدام از edge ها ظرفیت برابر با 1 اختصاص می دهیم. سپس الگوریتم ford-fulkerson را روی آن اجرا می کنیم. شار بیشینه ای که بدست می آید، در واقع همان تعداد مسیرهای یال مجزا در شبکه شار است. برای هر یال فقط یکبار قابل استفاده برای عبور شار است. پس تمام یال های استفاده شده در مسیرهای پیدا شده، مجزا هستند.

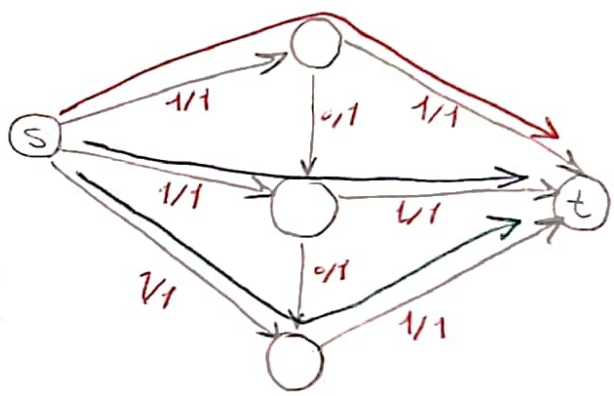
به طور مثال:

شبکه شار مقابل را داریم:



⇐ ظرفیت تمام یال ها را برابر 1 قرار دادیم.

حال الگوریتم ford-fulkerson را روی این شبکه اجرا می کنیم:



شار بیشینه در این شبکه، برابر 3 بدست آمد که همانطور که مشخص است، تعداد مسیرهای یال مجزا در آن هم برابر 3 تاست. ∅

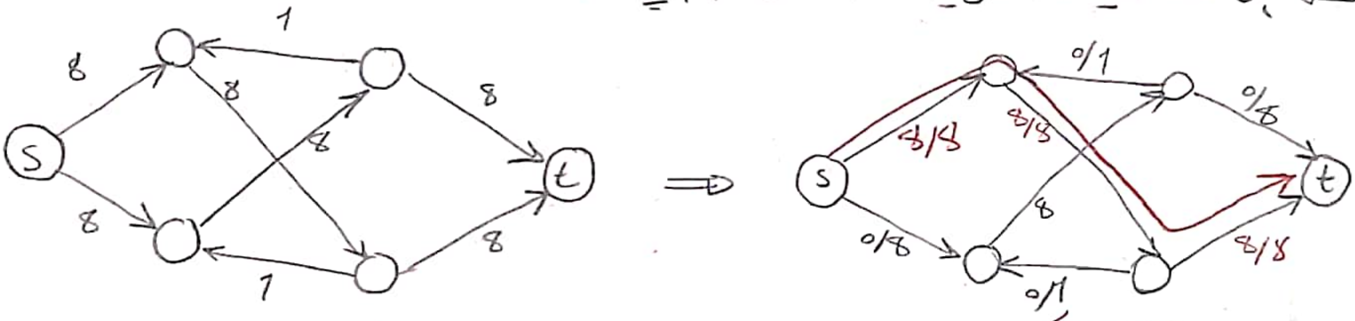
② در یک درخت با یال های بدون وزن، با cut کردن هر کدام از یال ها به minimum cut دست خواهیم یافت. چرا که با این کار، درخت (که همبند است) به دو بخش تقسیم خواهد شد. و از این جنبه دیگر هم قابل تقسیم نیست. بنابراین n-1 حالت برای برش کمینه، در یک درخت با n-1 یال خواهیم داشت.

پس تعداد برش های کمینه برابر n-1 است. ✓

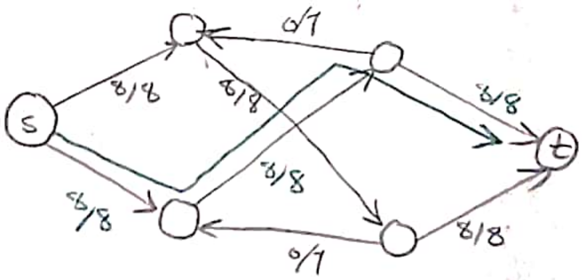
---

③ برای اینکه حداکثر تعداد عملیات افزایش شار را بدست بیاوریم، حالتی را در نظر می‌گیریم که الگوریتم ford-fulkerson در هر بار، مسیر با بیشترین ظرفیت (bottleneck بزرگتر) را به عنوان مسیر افزایشی انتخاب می‌کند. و طبق فرض مسئله، بیشترین flow ممکن (برابر با ظرفیت یال bottleneck) را عبور می‌دهد.

⇐ پس تعداد عملیات افزایش شار حداکثر برابر 2 است:



عملیات یک



⇐ بعد از این دو عملیات و عبور شار $8+8=16$ از s به t، دیگر در گراف، مسیر افزایشی وجود ندارد.

عملیات 2 ✓

④

+ گراف شبکه با یال های دارای ظرفیت بزرگتر از 1 داریم.

در این گراف، min cut ها را در نظر می‌گیریم. min cut با کمترین تعداد یال را C_1 می‌نامیم و تعداد یال هایش را برابر E_1 در نظر می‌گیریم. min cut با بیشترین تعداد یال را هم C_2 می‌نامیم و تعداد یال هایش را برابر E_2 در نظر می‌گیریم.

⊕ وقتی به ظرفیت هر یک از یال های گراف، یک واحد اضافه کنیم، به ظرفیت هر یک از min cut ها به اندازه‌ی تعداد یال های آن min cut اضافه می‌شود. در این صورت اگر C_1 را در نظر بگیریم، ظرفیتش برابر E_1 + (ظرفیت min cut قبل از افزایش ظرفیت یال ها) خواهد شد. و اگر C_2 را در نظر بگیریم، ظرفیتش برابر E_2 + (ظرفیت min cut قبل از افزایش ظرفیت یال ها) خواهد شد.

مشخص است که با توجه به اینکه $E_1 < E_2$ است، ظرفیت C_2 هم بیشتر از ظرفیت C_1 خواهد شد.

پس min cut در گراف جدید برابر C_1 خواهد بود. و max flow در گراف جدید، حداکثر به اندازه‌ی ظرفیت C_1 خواهد بود.

max flow پس از افزایش ظرفیت ها $\leqslant$ (ظرفیت min cut قبل از افزایش ظرفیت یال ها) $+ E_1 = m + E_1$

اندازه‌ی max flow قبل از افزایش ظرفیت یال ها $= m$

⊕ وقتی از ظرفیت هر یک از یال های گراف، یک واحد کم کنیم، از ظرفیت هر یک از min cut ها به اندازه‌ی تعداد یال های آن min cut کم می‌شود. در این صورت اگر C_1 را در نظر بگیریم، ظرفیتش برابر $m - E_1$ خواهد شد، و اگر C_2 را در نظر بگیریم، ظرفیتش برابر $m - E_2$ خواهد شد. مشخص است که با توجه به اینکه $E_1 < E_2$ است، ظرفیت C_1 بیشتر از ظرفیت C_2 خواهد شد. پس min cut در گراف جدید برابر C_2 خواهد بود. و max flow در گراف جدید، حداکثر به اندازه‌ی ظرفیت C_2 خواهد بود.

max flow پس از کاهش ظرفیت ها $\leqslant m - E_2$

ادامه ی ④

⊕ پس دیدیم درحالتی که یک واحد به ظرفیت همه ی یال ها اضافه می کنیم، حداکثر E_1 به جریان بیشینه اضافه می شود

پس $d_1 \leq E_1$ است.

⊕ وقتی که یک واحد از ظرفیت همه ی یال ها کم کنیم، حداقل E_2 از جریان بیشینه کم می شود. پس

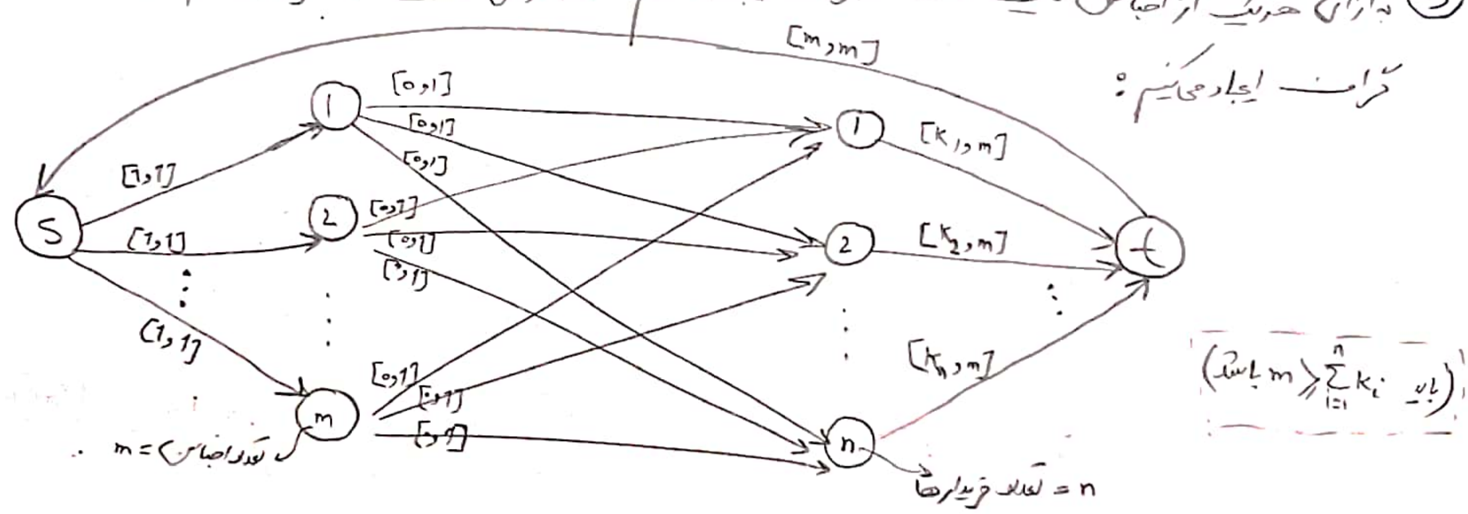
$d_2 \geq E_2$ است.

← با توجه به اینکه داشتیم $E_1 < E_2$، داریم: $d_1 \leq E_1 \leq E_2 \leq d_2$ ⟸ پس اثبات شد:

$(d_1 \leq d_2)$ ✓

---

⑤ به ازای هریک از اجناس، یک node در گراف ایجاد می کنیم. و به ازای هریک از خریدارها هم یک node در گراف ایجاد می کنیم:



← به ازای هریک از گره های متناظر اجناس، یک یال از آن گره به همه ی گره های مربوط به خریدارها وصل می کنیم.

← به ازای هر یال موجود در گراف، یک شرط $[x, y]$ داریم که نشان می دهد flow عبوری از آن یال باید حداقل x و حداکثر y باشد.

← برای هرکدام از یال های خروجی از اجناس و ورودی به خریدارها، شرط $[0, 1]$ را قرار می دهیم. به این معنا که هر خریدار، حداقل 0 عدد از یک جنس خاص و حداکثر 1 عدد از آن جنس را خریداری می کند.

ادامه‌ی ⑤

سپس دو گره s و t به عنوان super-source و super-sink در نظر می‌گیریم.

سپس از گره s به هر یک از node های نظیر اجناس، یک یال با شرط [1,1] رسم می‌کنیم. به این معنی که حداقل 1 و حداکثر 1 عدد از هر جنس تولید شده و وجود دارد. (دقیقاً 1 عدد)

سپس از هر گره نظیر خریدارها، یک یال با شرط $[k_i, m]$ به t، رسم می‌کنیم. به این معنی که هر خریدار را در نهایت حداقل k_i و حداکثر m تا جنس خریداری کرده است. به این دلیل شرط حداکثر m را قرار دادیم، چون در کل m تا جنس داریم و در حداکثر حالت، تعداد خریدهای یک خریدار، یک خریدار در کل می‌توانیم داشته باشیم که تمام m تا جنس را خریداری کند.

سپس یک یال از t به s با شرط [m, m] داریم.

سپس مسئله‌ی مورد نظر ما در این سوال، به مسئله‌ی circulation تبدیل شد.

⊕ باید اثبات کنیم: ① وقتی یک circulation در این گراف داریم ⟸ یک تقسیم و فروش satisfiable برای اجناس وجود دارد.
② وقتی یک تقسیم و فروش satisfiable برای اجناس وجود دارد ⟸ یک circulation در این گراف داریم.

① وقتی یک circulation با سایز m داریم، می‌توانیم از s به هر کدام از node های نظیر اجناس، یک flow با سایز 1 وارد کنیم. که این اجناس هم m تا هستند. با توجه به اینکه ظرفیت‌ها همگی integer هستند، flow با سایز 1 که از هر کدام از اجناس خارج می‌شود، دقیقاً به یکی از node های مربوط به خریدارها وارد می‌شود. پس هر یک اجناس دقیقاً به یک خریدار فروخته می‌شود. و flow هایی که به هر یک از node های مربوط به خریدارها وارد می‌شود، حداقل k_i و حداکثر m واحد است. پس شرط حداقل k_i جنس برای خریدار i ام هم برقرار است ⟸ پس یک تقسیم و فروش valid برای اجناس بین خریدارها برقرار شده.

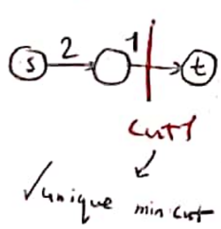
② وقتی یک تقسیم و فروش valid برای اجناس داشته باشیم، یک circulation با سایز m به این شکل ایجاد می‌کنیم:
ابتدا به ازای هر کدام از یال‌های بین s و node های نظیر اجناس، یک flow با سایز 1 عبور می‌دهیم. که این node ها m تا هستند. با توجه به تقسیم valid، flow ورودی به هر جنس را به خریدار مورد نظر در تقسیم valid، وارد می‌کنیم. هر کدام از خریدارها حداقل k_i و حداکثر m تا جنس خریده است. پس flow به این اندازه به آن وارد می‌شود و از آن خارج می‌شود و به t می‌رود. در نهایت یک flow با سایز m از t به s می‌رود. ⟸ پس یک circulation با سایز m در گراف برقرار شده.

ابتدا الگوریتم ford-fulkerson را روی این گراف اجرا می‌کنیم و max flow را بدست می‌آوریم. به همان اندازه‌ی min cut در این گراف است. سپس با توجه به اینکه اندازه‌ی min cut را در اختیار داریم، یک min cut را پیدا می‌کنیم. یال‌های این min cut را در نظر می‌گیریم. هربار یکی از این یال‌ها را در نظر می‌گیریم و ظرفیت آن را به اندازه‌ی 1 واحد افزایش می‌دهیم. اگر به ازای افزایش یک واحدی ظرفیت حداقل یکی از این یال‌ها، اندازه‌ی max flow در این گراف تغییر نکند، یعنی در مسیر ای که از آن یال flow می‌گذرد، این یال، bottleneck آن مسیر نبوده است. پس یک min cut دیگر در این گراف وجود دارد، که یکی از یال‌هایش، bottleneck مربوط به این مسیر است. اما اگر به ازای همه‌ی یال‌های min cut، هربار ظرفیت یکی از یال‌ها را یک واحد افزایش دادیم و هربار اندازه‌ی max flow (یعنی همان min cut) گراف جدید، نسبت به اندازه‌ی min cut اولیه افزایش یافت، یعنی این min cut یافت شده، تنها min cut موجود در این گراف است. و unique است. و سایر cut های موجود در این گراف، همگی اکیداً بزرگ‌تر از این min cut مورد نظر است.

← در این الگوریتم، یک بار در ابتدا برای یافتن min cut در گراف اصلی، ford-fulkerson را اجرا می‌کنیم. و سپس به تعداد یال‌های شرکت کننده در min cut یافت شده، ford-fulkerson را مجدداً به شرح بالا اجرا می‌کنیم. اگر تعداد یال‌های min cut را برابر K فرض کنیم، هزینه‌ی الگوریتم، به شرح زیر است:

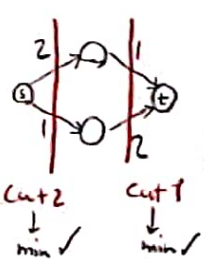
$$(k+1) \times \underbrace{O(V.E^2)}_{\text{ford-fulkerson هزینه‌ی}} \longrightarrow \text{polynomial}$$

← مثال گراف با unique min cut:

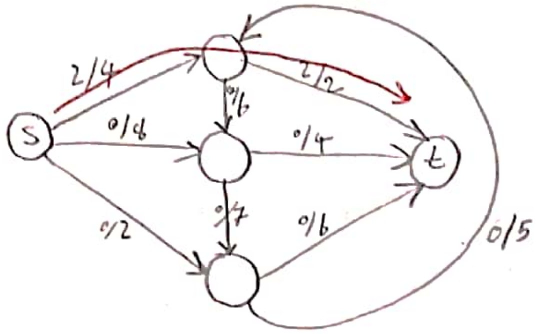


اگر ظرفیت تنها یال cut1 را یک واحد افزایش دهیم، max flow هم یک واحد زیاد می‌شود و برابر 2 می‌شود.

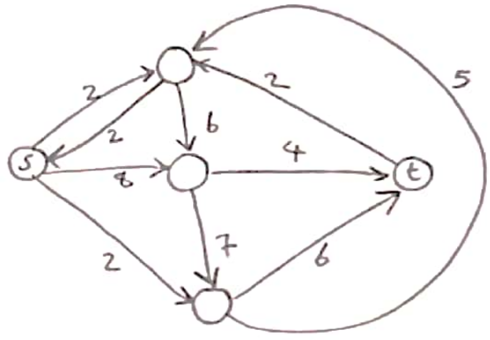
← مثال گراف با چند min cut:



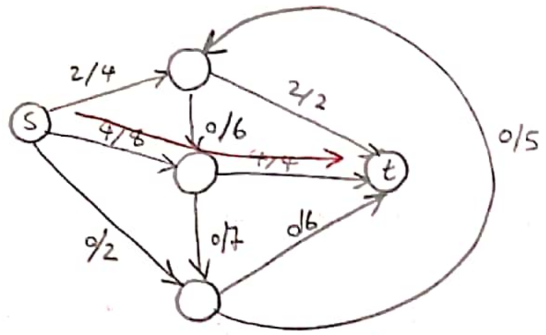
اگر در cut1، ظرفیت یال‌ها یکی را برابر 3 کنیم، max flow تغییر نمی‌کند و همچنان برابر 2 خواهد ماند.



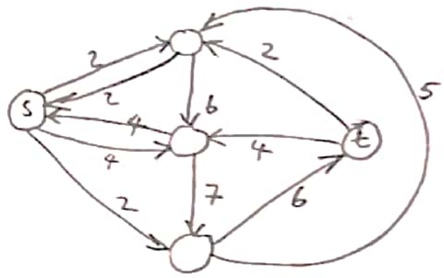
$G_1 \longrightarrow flow = 2$



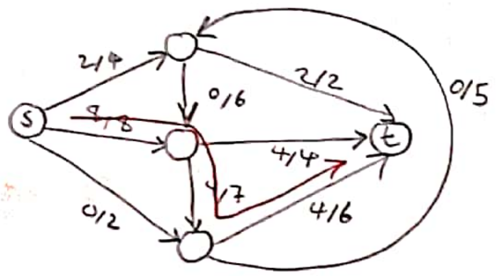
G_{f_1}



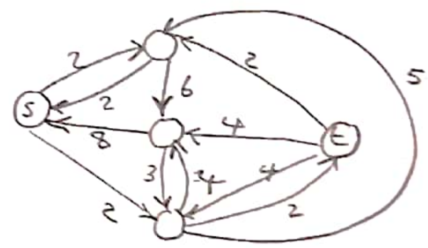
$G_2 \longrightarrow flow = 2+4 = 6$



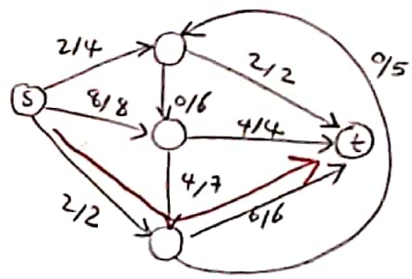
G_{f_2}



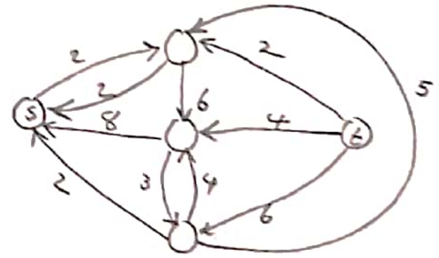
$G_3 \longrightarrow flow = 6+4 = 10$



G_{f_3}



$G_4 \longrightarrow flow = 10+2 = 12$



G_{f_4}

که هیچ مسیر افزایشی دیگری در این گراف باقی مانده وجود ندارد. پس max flow برابر 12 است.