طراحی الگوریتم‌ها

بهار ۱۴۰۰

مدرس: مسعود صدیقین

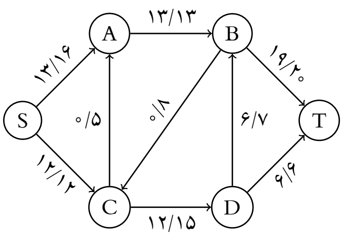
دانشکده‌ی مهندسی کامپیوتر

یادآوری جلسه هفدهم — **شار بیشینه** — گردآورنده: دانیال فراهانی ــ زهرا فاضل

در جلسه گذشته با مسئله شار بیشینه آشنا شدیم و دو الگوریتم برای حل آن ارائه دادیم.

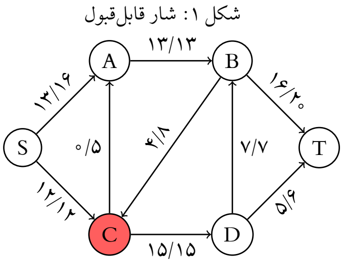
مسئله شار بیشینه

گراف وزن‌دار و جهت‌دار G و دو رأس مبدأ و مقصد s و t را داریم. وزن هر یال نشان‌دهنده ظرفیت آن یال است. می‌خواهیم بیشینه شار از s به t را بیابیم.



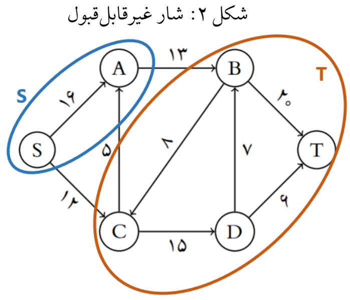
شکل ۱: شار قابل‌قبول

تعریف ۱. شار بیشینه بین s و t برابر با مقدار وزنی که از مسیرهای بین این دو رأس می‌توان جابه‌جا کرد به‌گونه‌ای که هیچ یالی بیشتر از ظرفیتش شار حمل نکند و به جز رأس مبدأ و مقصد، شار ورودی به هر رأس با شار خروجی آن برابر باشد. به عنوان مثال شار شبکه ۱ قابل‌قبول و شار شبکه ۲ غیرقابل‌قبول است.



شکل ۲: شار غیرقابل‌قبول

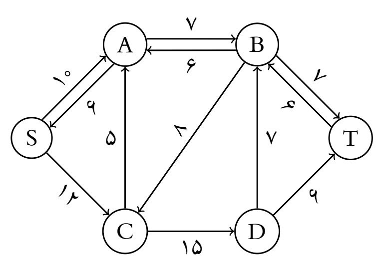
تعریف ۲. برش S و T یک تقسیم رئوس گراف به دو مجموعه S و T است به صورتی که $s \in S$ و $t \in T$. ظرفیت برش برابر با مجموع وزن یال‌های از S به T است.



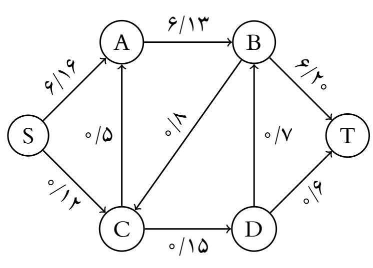
تعریف ۳. برش کمینه، برش با ظرفیت کمینه است. شکل روبه‌رو برش کمینه شبکه داده‌شده را نشان می‌دهد.

قضیه ۱. شار بیشینه برابر با برش کمینه است.

تعریف ۴. گراف باقی‌مانده گرافی با مجموعه رئوس گراف اصلی است که به ازای هر یال در گراف اصلی دو یال در آن قرار می‌دهیم، به‌گونه‌ای که اگر در گراف اصلی از رأس A به رأس B یک یال با ظرفیت c و شار f داشته‌باشیم، در این گراف یک یال از A به B با ظرفیت c-f و یک یال از B به A با ظرفیت f داریم. به عنوان مثال گراف باقی‌مانده گراف ۴ مطابق شکل ۳ است.



شکل ۳: گراف باقی‌مانده

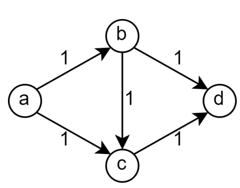


شکل ۴: گراف اصلی

برای حل این مسئله ۳ الگوریتم ارائه دادیم که در ادامه آن‌ها را بررسی می‌کنیم.

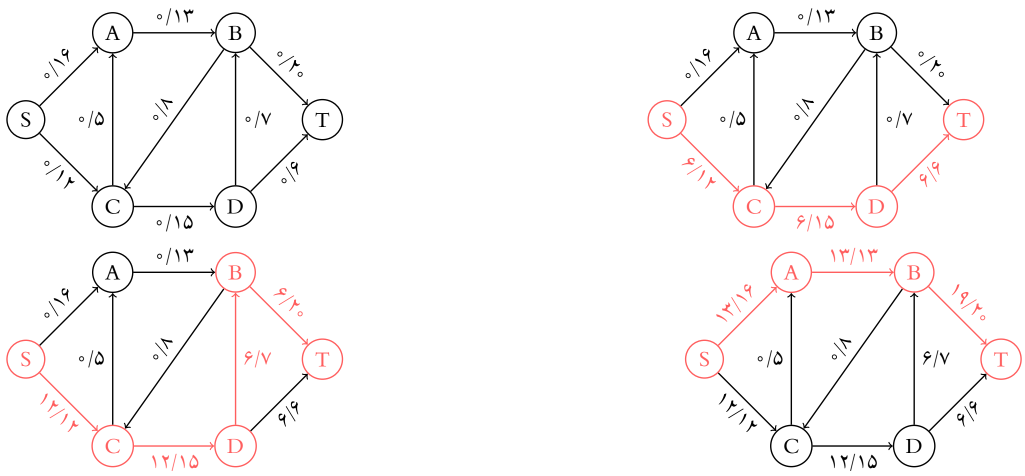
الگوریتم حریصانه

الگوریتم حریصانه: با شار ۰ به ازای هر یال شروع می‌کنیم. از روی گراف G گراف G_f را می‌سازیم که شار باقی‌مانده هر یال را نشان دهد و یال‌های با ظرفیت ۰ را حذف می‌کنیم. در G_f مسیری از s به t پیدا می‌کنیم. شار یالهای این مسیر را به اندازه کمینه ظرفیت مسیر زیاد می‌کنیم. این الگوریتم صحیح نیست زیرا برای شبکه روبه‌رو خروجی آن ۱ است در حالی که شار بیشینه ۲ است.



الگوریتم فورد_ فولکرسون

برای حل مشکل الگوریتم قبلی نیاز داریم که بتوانیم شار هر یال را کم کنیم. برای این کار از گراف باقی‌مانده استفاده می‌کنیم. الگوریتم ۱ را به این صورت اجرا می‌کنیم که در هر مرحله مسیری از s به t در گراف باقی‌مانده پیدا کرده و شار یال‌ها را به‌روزرسانی می‌کنیم. زمان اجرای این الگوریتم بستگی به تعداد دفعات اجرای DFS دارد و آن هم وابسته به بیشینه وزن یال‌هاست. بنابراین الگوریتم الزاما سریع نخواهدبود. در ادامه مثالی از اجرای این الگوریتم آمده‌است.



الگوریتم ادموندز_ کارپ

مشابه الگوریتم فورد_ فولکرسون است با این تفاوت که برای مسیریابی از الگوریتم BFS به جای DFS استفاده می‌کنیم. این کار باعث می‌شود که زمان اجرا مستقل از بیشینه وزن یال‌ها باشد و زمان اجرا $O(nm^2)$ خواهدبود.

برای اثبات درستی الگوریتم از قضیه زیر استفاده می‌کنیم:

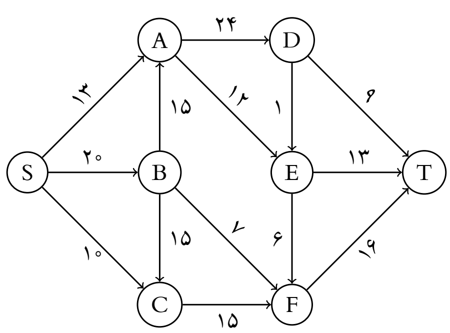
قضیه ۲. فرض کنید G یک شبکه شار و f یک شار از G باشد. آنگاه گزاره‌های زیر معادل‌اند:

۱. f شار بیشینه است.

۲. G_f هیچ مسیری از s به t ندارد.

۳. برش S و T در G وجود دارد که $c(S,T) = |f^*|$.

پرسش در شبکه شار زیر، شار بیشینه و برش کمینه را بیابید.



پاسخ های خود را می توانید تا قبل از شروع کلاس به این لینک ارسال کنید.