دانشکده‌ی علوم ریاضی

آنالیز الگوریتم — ۱۸ فروردین ۹۲

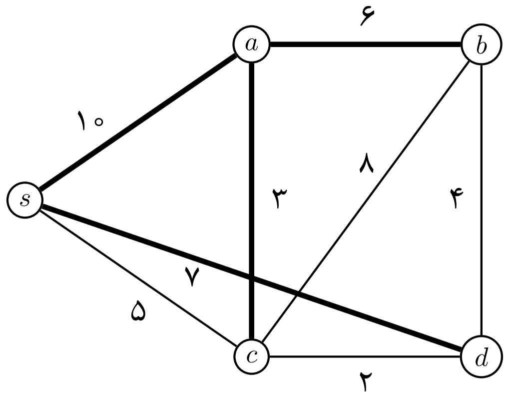
جلسه‌ی ۱۱: درخت فراگیر کمینه در گراف

مدرّس: دکتر شهرام خزائی — نگارنده: افشین زارعی

۱ تعاریف

تعریف ۱ *(درخت فراگیر) زیرگراف $G' = (V, E')$ یک درخت فراگیر برای گراف $G = (V, E)$ است، اگر و فقط اگر G' یک درخت باشد.*

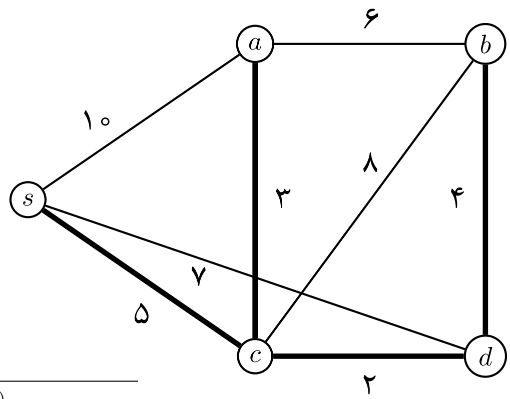
مثال ۱ *در گراف زیر یال‌های مشخص شده، تشکیل یک درخت فراگیر برای گراف می‌دهند.*



نکته ۱ *در گراف ناهمبند درخت فراگیر وجود ندارد.*

تعریف ۲ *(درخت فراگیر کمینه[1]) درخت فراگیر T را درخت فراگیر کمینه گراف وزن‌دار G گوییم، هرگاه مجموع وزن یال‌های آن نسبت به بقیه‌ی درخت‌های فراگیر کمترین مقدار باشد.*

مثال ۲ *در گراف زیر یال‌های مشخص شده، درخت فراگیر کمینه گراف است.*



[1] Minimum Spanning Trees (MST)

۲ پیدا کردن درخت فراگیر کمینه

فرض کنید یک گراف بدون جهت همبند $G=(V,E)$ و وزن‌های[۲] c_e برای هر یال $e\in E$ و نیز نمایش گراف بصورت لیست مجاورت را داریم، مسأله‌ی ما پیدا کردن درخت فراگیر کمینه این گراف است.
بعنوان مثال فرض کنید که یک طراح شهری می‌خواهد چند شهر را با جاده‌هایی به هم متصل کند، به طوری که امکان حرکت از هر شهری به شهر دیگر وجود داشته باشد. اگر برای طراحی این جاده‌ها بودجه محدود پیش بینی شده باشد، او احتمالا در طراحی خود حداقل طول کل جاده‌ها را درنظر می‌گیرد.
هدف مسأله درخت فراگیر کمینه، ارائه الگوریتمی برای حل چنین مسائلی می‌باشد.
ما در این جلسه فرض می‌کنیم که همه‌ی وزن‌ها متمایز متمایزند و نتایج خود را با این فرض ارائه می‌دهیم که تحت آن می‌توان نشان داد درخت فراگیر کمینه یکتاست. اما نتایج در حالت کلی نیز صادق است و تعمیم آن‌ها به خواننده واگذار می‌شود.

لم ۱ *اگر در گراف همبند همه‌ی وزن‌ها متمایز باشند، درخت فراگیر کمینه یکتاست.*

۳ الگوریتم‌های حل مسأله

دو الگوریتم را ما برای حل مسأله بررسی می‌کنیم.

۱. الگوریتم پریم[۳]، این الگوریتم را رابرت سی پریم[۴] در سال ۱۹۵۷ ارائه کرد، ولی قبلاً توسط یارنیک[۵] در سال ۱۹۳۰ کشف شده بود.

۲. الگوریتم کروسکال[۶]، که در سال ۱۹۵۶ کشف شد.

۱.۳ الگوریتم پریم

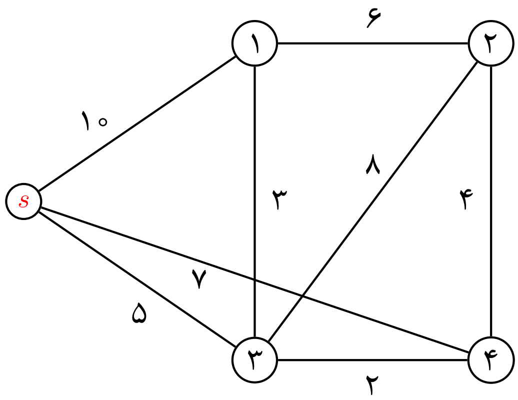
در این الگوریتم ما ابتدا یک مجموعه‌ی X را که شامل یک رأس دلخواه s است، در نظر می‌گیریم و سپس از بین تمام یال‌هایی که یک سر آن‌ها در مجموعه‌ی X قرار دارد و سر دیگر آن‌ها در مجموعه‌ی $V-X$ است، یالی را که دارای کمترین وزن است، انتخاب می‌کنیم و آن یال را به درخت فراگیر و رأس مربوط به آن را به مجموعه‌ی X می‌افزاییم. حال این اعمال را برروی مجموعه‌ی جدید X تا زمانی که مجموعه‌ی X همه‌ی رئوس گراف را در برگیرد، انجام می‌دهیم. شبه کد این الگوریتم بصورت زیر است.

Algorithm 1 Algorithm: PRIM-MST

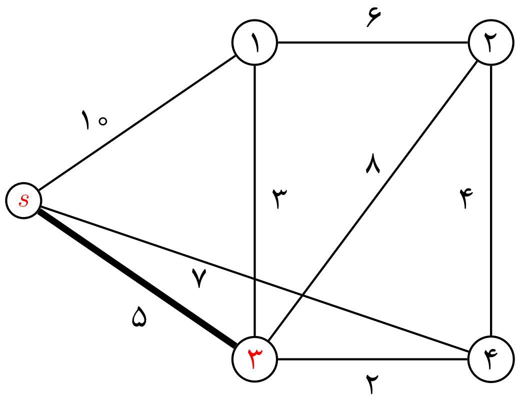
```
function PRIM-MST(graph G = (V, E) , costs {c_e}_{e∈E} )
    T = φ
    X = {s} for some s ∈ V
    while X ≠ V do
        Let e = (u, v) be the cheapest edge of G with u ∈ X and v ∈ V − X
        add e to T
        add v to X
    return T
```

---
[۲] weights/costs
[۳] Prim's MST Algorithm
[۴] Robert C. Prim
[۵] Jarnik
[۶] Kruskal's algorithm

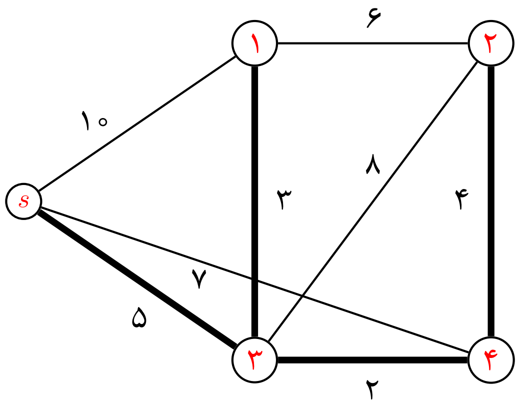
مثال ۳ *الگوریتم پریم را بر روی گراف زیر اعمال می‌کنیم.*



پس از یک مرحله اجرای الگوریتم پریم با رأس اولیه‌ی s، یالی که مشخص شده، را به درخت فراگیر کمینه و رأس متناظر را به مجموعه‌ی X می‌افزاید.



پس از پایان الگوریتم پریم، خروجی بصورت زیر است. که همان درخت فراگیر کمینه گراف است.



۲.۳ صحت

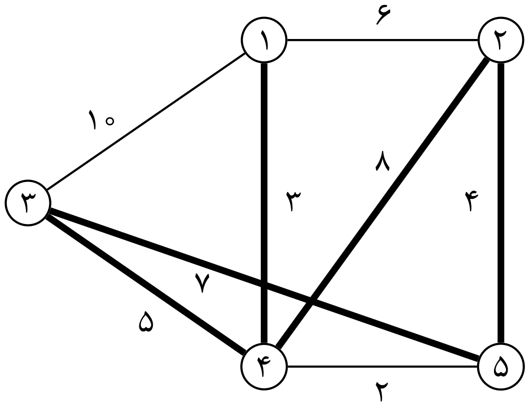
برای اثبات صحت این الگوریتم ابتدا چند تعریف و لم ارائه می‌دهیم و سپس با استفاده از آن‌ها درستی الگوریتم را ثابت می‌کنیم.

تعریف ۳ (*برش*[۷]) *دوتایی $\langle A, B\rangle$ یک برش از گراف $G = (V, E)$ است، اگر مجموعه‌های A, B مجموعه‌ی V را افراز کنند.*

تعریف ۴ (*یال برشی*[۸]) *هر یال که یک سر آن در A و سر دیگر آن در B باشد، یک یال برشی برای برش $\langle A, B\rangle$ نامیده می‌شود.*

[۷] cut
[۸] crossing edge

مثال ۴ *در گراف زیر مجموعه‌های $A = \{۱, ۲, ۳\}$ و $B = \{۴, ۵\}$ یک برش برای گراف است و یال‌های مشخص شده، یال‌های برشی این برش هستند.*



لم ۲ *یک گراف ناهمبند است اگر و فقط اگر دارای یک برش بدون یال برشی باشد.*

برهان. ($\Rightarrow$) اگر $\langle A, B\rangle$ یک برش بدون یال برشی برای گراف G باشد، نشان می‌دهیم گراف G ناهمبند است. برای این کار نشان می‌دهیم گره‌های u و v وجود دارند که مسیری از u به v در گراف G وجود ندارد. گره‌های دلخواه $u \in A$ و $v \in B$ را در نظر بگیرید. در این صورت نشان می‌دهیم هیچ مسیری از u به v در گراف G وجود ندارد، پس گراف G ناهمبند است. برای نشان دادن این موضوع فرض خلف کنید که مسیری چون P از u به v در گراف G وجود داشته باشد. فرض کنید که v' اولین رأسی در مسیر P باشد که در B قرار دارد (چنین رأسی وجود دارد چون $v \in B$ و $v \in P$) و رأسی که دقیقاً قبل از v' در مسیر P قرار دارد را با u' نشان دهید (یعنی $u' \in A$ رأسی باشد که مسیر P بوسیله‌ی آن از A خارج می‌شود). در این صورت $u'v'$ یک یال برشی برای برش $\langle A, B\rangle$ است که تناقض است.
($\Leftarrow$) یک رأس دلخواه u درنظر گرفته، و تمام رئوس قابل دسترسی از آن را در A گذاشته و بقیه را در B می‌گذاریم، بوضوح $\langle A, B\rangle$ یک برش بدون یال برشی برای گراف هستند.

■

لم ۳ *اگر یک یال برشی متعلق به یک دور باشد، حداقل یک یال برشی دیگر متعلق به آن دور وجود دارد.*

حکم قوی‌تر از لم فوق نیز صادق است بدین صورت که تعداد یال‌های برشی هر دور زوج است.

نتیجه ۱ *اگر یالی تنها یال برشی باشد، آن یال متعلق به هیچ دوری نیست.*

تعریف ۵ *(ویژگی برش[۹]) گوییم یال e دارای ویژگی برش است، اگر یال برشی با کمترین وزن برای یک برش $\langle A, B\rangle$ باشد.*

مثال ۵ *در گراف مثال ۴ یال $(۱, ۴)$ یال با ویژگی برش برای آن برش است.*

لم ۴ *اگر یالی را به یک درخت اضافه کنیم، که در آن درخت نیست، باعث ایجاد یک دور در درخت می‌شود، و اگر از همان دور به‌دست آمده یک یال را حذف کنیم، حاصل یک درخت می‌شود.*

لم ۵ *تحت فرض متمایز بودن وزن‌ها، هر یال دارای ویژگی برش، متعلق به تنها درخت فراگیر کمینه است.*

برهان. فرض کنید که T درخت فراگیر کمینه باشد. با استفاده از برهان خلف لم را اثبات می‌کنیم. فرض خلف: فرض کنید $e = (u, v)$ یک یال برشی با ویژگی برش برای برش $\langle S, V - S\rangle$ باشد که در درخت فراگیر کمینه‌ی T قرار ندارد. بدون کاستن از کلیت مسأله فرض کنید $u \in S$، $v \in V - S$. چون T درخت فراگیر است، بنابراین یک مسیر یکتای P از u به v در T وجود دارد. فرض کنید که v' اولین رأس مسیر P باشد که متعلق به مجموعه‌ی $V - S$ است و نیز فرض کنید که u' رأسی باشد که دقیقاً قبل از v' قرار دارد ($u' \in S$ رأسی باشد که مسیر P بوسیله‌ی آن از S خارج می‌شود) و فرض کنید که $e' = (u', v')$. پس e' یال برشی برش $\langle S, V - S\rangle$ است. حال $T' = (T - \{e'\}) \bigcup \{e\}$ را در نظر بگیرید. T' نیزیک درخت فراگیر است (همه‌ی رئوس را داراست). از طرفی چون e دارای ویژگی برش است، پس $c_e < c_{e'}$ و چون دو درخت فراگیر T, T' فقط در این یال با هم تفاوت دارند، بنابراین $C(T') < C(T)$. که این تناقض است.

■

[۹]Cut property

قضیه ۶ *الگوریتم پریم همواره یک درخت فراگیر کمینه را محاسبه می‌کند.*

برای اثبات قضیه، ابتدا ثابت می‌کنیم که خروجی الگوریتم یک درخت فراگیر است و سپس کمینه بودن آن را ثابت می‌کنیم.

لم ۷ *خروجی الگوریتم یک درخت فراگیر است.*

برهان. ثابت می‌کنیم که (X,T) بعد از هر تکرار یک درخت است و الگوریتم وقتی خاتمه می‌یابد که $X=V$.
(X,T) همبند است؛ زیرا هر گره جدید که می‌افزاییم با یالی به گره‌های قبلی متصل می‌شود، پس همبندی حفظ می‌شود. از طرفی (X,T) دور ندارد؛ چون آخرین یال اضافه شده پس از هر مرحله از الگوریتم پریم تنها یال برشی برش $(X, V-X)$ از گراف (V,T) است، پس این گراف دور ندارد و با توجه به اینکه (X,T) زیرگراف (V,T) است، آن هم دور ندارد. به وضوح نیز حلقه‌ی **while** زمانی به اتمام می‌رسد، که $X=V$. ■

لم ۸ *درخت فراگیر خروجی، کمینه است.*

برهان. فرض کنید خروجی الگوریتم پریم درخت T^* باشد (طبق لم ۲ درخت است). پس $C(MST) \leq C(T^*)$. پس کافی است نشان بدهیم $C(T^*) \leq C(MST)$. چون در هر مرحله از الگوریتم پریم یالی که اضافه می‌شود، دارای ویژگی برش است، پس طبق لم ۳ متعلق به MST است. بنابراین هر یالی که در T^* قرار دارد در MST نیز قرار دارد، پس $C(T^*) \leq C(MST)$. پس خروجی الگوریتم درخت فراگیر کمینه است. ■

۳.۳ پیچیدگی الگوریتم

چون حلقه‌ی **while** در $O(n)$ انجام می‌شود و در هر تکرار در بدترین حالت تمام یال‌ها را باید بررسی کند، پس پیچیدگی این الگوریتم برابر $O(mn)$ است.
این پیچیدگی را می‌توان با استفاده از داده ساختار هرم[۱۰] به $O(m\log n)$ کاهش داد. بدین صورت که اعضای هرم رئوس هستند و مقادیر کلید وزن‌های یال‌ها است.

[۱۰] heap