جلسه بیست و ششم

13:17 Sunday, December 26, 2021

موضوع : درخت پوشای کمینه .

درخت : گراف همبند بدون دور

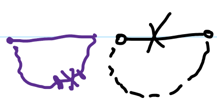
۱- هر درخت دارای حداقل ۲ برگ است : طولانی‌ترین مسیر

۲- هر درخت n راسی دارای $n-1$ یال است . استقرا

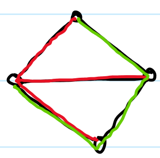
۳- بین هر ۲ راس دقیقاً یک مسیر وجود دارد :

۴- اضافه کردن هر یال به درخت ایجاد دوری می‌کند . این دور شامل یال اضافه شده است

۵- حذف هر یال از درخت آن را ناهمبند می‌کند



* زیر درخت فراگیر یک گراف : زیرگرافی شامل همه راس های آن گراف است که درخت باشد .



قضیه کیلی : تعداد زیر درخت‌های فراگیر گراف کامل n راسی ؟ n^{n-2}

(راس ها : $1-n$)

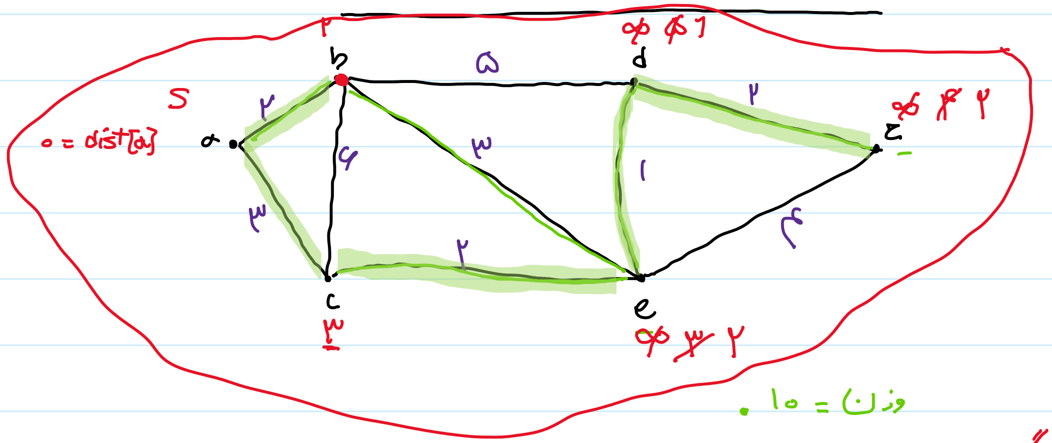
همبند ـ بدون جهت ـ وزن دار .

مسئله : گراف وزن دار G داده شده است . هدف پیدا کردن یک زیر درخت فراگیر T از G است که مجموع وزن یال های آن کمینه شود .

MST . دیپک

الگوریتم Prim پریم

الگوریتم Kraskal کراسکال



وزن = ۱۰

الگوریتم پریم :

$S = \{a\}$

$dist[b] = \begin{cases} \infty & \text{اگر a و b متصل نباشند} \\ \omega(a, b) & \text{اگر a و b متصل باشند} \end{cases}$

* در هر مرحله، راسی که مقدار dist آن کمینه است و در V/S قرار دارد را به S اضافه می‌کنیم. یالی که آن مقدار dist را برای آن راس ایجاد کرد را به درخت پوشای کمینه اضافه می‌کنیم. مقدار dist سایر راس‌ها را به روز می‌کنیم.

$S = \{v_1\}$

$dist\{v_j\} = \begin{cases} \infty & \text{اگر یال نباشد} \\ \omega(v_i, v_j) & \text{اگر یال باشد} \end{cases}$

$father[v_j] \rightarrow$ راسی که با وزن کمینه به v_j متصل است (از S)

```
        while (S ≠ V) {
log n       select vj ∈ V/S with min dist
1           add vj to S
1           add edge (vj, father[vj]) to mst.
m log n     for every vk ∈ N(vj)
                if( vk ∉ S ∧ ω(vj, vk) < dist[vk]){
                    dist[vk] = ω(vj, vk)
                    father[vk] = vj
                }
        }
```

زمان اجرا : ماتریس مجاورت : $O(n^2)$

$O(n^2)$

زمان اجرا: ماتریس مجاورت : $O(n^2)$

لیست مجاورت : $O(n^2)$

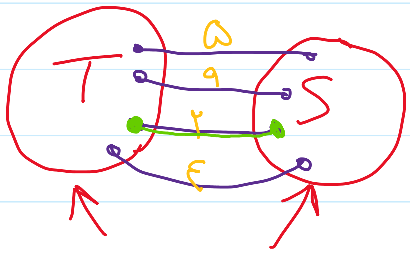
هرم + لیست مجاورت : $O(n\log n + m\log n)$

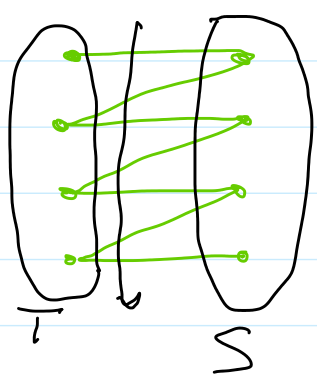
هرم فیبوناچی + 〃 : $O(m + n\log n)$

اثبات درستی الگوریتم پریم:

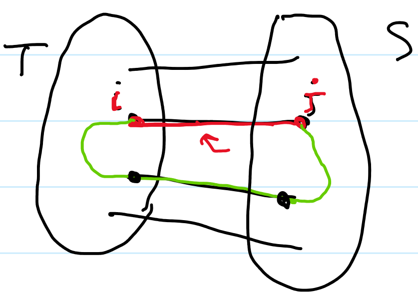
تعریف: فرض کنید راس های گراف G را به دو مجموعه S و T تقسیم کرده ایم. در این صورت، یال با وزن کمینه که یک سر آن در S و یک سر آن در T است حتماً در د پ ک وجود دارد.

MST





اثبات:



MST ↓

حداقل ۱ یال از مجموعه یال های بین S و T را استفاده کرده

دلیل

ناوز را اضافه کنیم ← ۱ دور ایجاد می شود ← یال این دور که بین S و T است را حذف می کنیم ← MST با وزن کمتر داریم ← تناقض!

چرا قضیه بالا درستی الگوریتم پریم را اثبات می کند؟

Remark : یال های منفی مهم نیست .