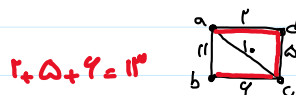


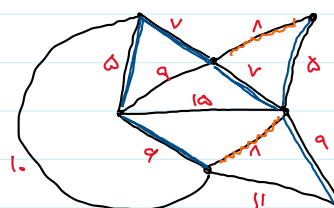
موضوع: درخت پوشای کمینه ... الگوریتم کراسکال



$$2 + 5 + 9 = 16$$

الگوریتم کراسکال: با $T = \emptyset$ شروع کن، هر مرحله یال با وزن کمینه را اضافه کردن آن ایجاد دور نمی کند یا به T اضافه کن.

← مرحله پنجم



* کامیار

اثبات درستی الگوریتم: برعکس خلف $OPT \neq ALG$

$$\rightarrow OPT: e_1, e_2, e_3, \dots, e_{n-1}$$

$$\rightarrow ALG: e'_1, e'_2, e'_3, \dots, e'_{n-1}$$

$$\omega(e_i) > \omega(e'_i)$$

فرض کنید i اولین اندیسی باشد که $e_i \neq e'_i$.

$$e'_1 = e_1, e'_2 = e_2, \dots, e'_{i-1} = e_{i-1}$$

در این مرحله اول کراسکال e'_i تا e'_{i-1} اضافه شدند.

در مرحله i ام: e'_i اضافه شده: e'_i کوچکترین یالی است که اضافه کردن آن در

مرحله i ام ایجاد دور نمی کرده است

چون e_i در OPT هست و OPT درخت است، در نتیجه e_i هم با e'_1, \dots, e'_{i-1}

ایجاد دور نمی کند.

$$\omega(e'_i) \leq \omega(e_i)$$

$$OPT: e_1, e_2, \dots, e_i, \dots, e_{n-1}$$

$$ALG: e'_1, e'_2, \dots, e'_i, \dots, e'_{n-1}$$

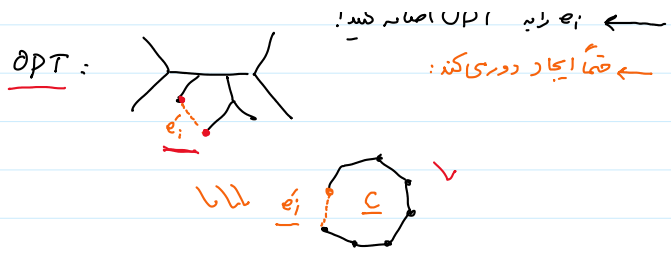
$$\omega(e'_i) < \omega(e_i)$$

← e'_i را به OPT اضافه کند!

← قضا ایجاد دور نمی کند:

OPT :





یک یال e_j در C وجود دارد که در Alg نیست.

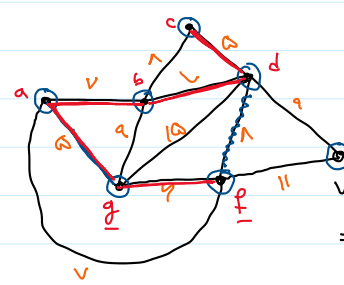
$$\omega(e_i) \leq \omega(e_j)$$

تا اولین اندیس نابرابر بود $\leftarrow i > j \leftarrow \omega(e_i) < \omega(e_j)$

e_i رابه OPT اضافه می کنیم و e_j را از OPT حذف می کنیم.
 ← وزن درخت هم می شود ← با همینه بودن OPT در تناقض.

پایان اثبات

زمان اجرای الگوریتم کراسکل:



* مرتب کردن یال: $\vee O(E \log E)$
 $= O(E \log n) \quad (\log E = \log n)$
 چک کردن دور:

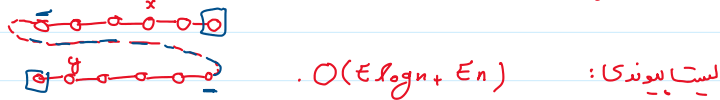
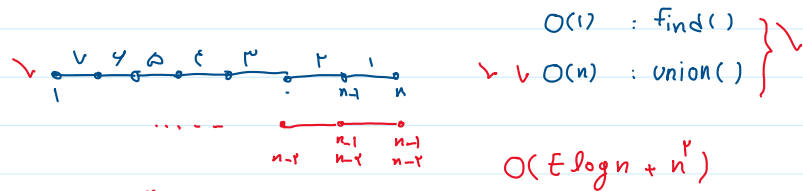
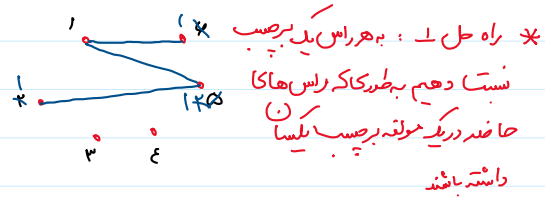
در ابتدا که $T = \emptyset$ ، n مولفه همبندی مختلف داریم.

چرا (g, a) دور ایجاد نمی کند؟ چون a و g به دو مولفه همبندی مختلف اختصاص دارند.
 چرا (f, d) را نمی توانیم؟ چون d و f به یک مولفه همبندی اختصاص دارند.

* یال (x, y) را می توان اضافه کرد $\Leftrightarrow x$ و y به دو مولفه همبندی مختلف تعلق داشته باشند

۱. * یال ها را مرتب کن $\leftarrow O(E \log n)$
 ۲. * به الای هوای نام، مولفه مربوط به \downarrow سریال را چک کن \leftarrow find $\leftarrow O(1)$
 اگر یکی بودند: اضافه کردنش ایجاد دوری کند
 نه نبذند: یال را اضافه کنند به T و مولفه ها دوسریال را \leftarrow union $\leftarrow O(n)$
 ترکیب کنند.





بهبود نیست!
 * بهیچ راه حل 1) همیشه برچسب مجموعه کچله را عوض کنیم: اگر برچسب یک راس عوض شود، اندازه مولفه آن حداقل 2 برابر می شود
 ⇐ حداقل $\log n$ بار برچسب عوض می شود
 ⇐ تعداد کل تغییر برچسب ها $O(n \log n)$
 ⇐ زمان کل: $O(E \log n) = O(E \log n + n \log n)$

$$O(E \log n + n^2) = O(E \log n + E + n^2)$$

داده ساختار Union-Find : [disjointset]
 $\begin{cases} \text{Find} = O(\log^* n) \\ \text{Union} = O(\log^* n) \end{cases}$
 $\log^* 2 = 1, \log^* 2^2 = 2, \log^* 2^{2^2} = 3, \log^* 2^{2^{2^2}} = 4, 2^{16} = 65536$
 $\log^*(?) = 5 \Rightarrow ? \approx 65536$

زمان اجرای کراسکل:
 $O(E \log n + E \log^* n + n \log^* n) = O(E \log n)$
 * اگر به ترتیب سازی خطی انجام دهیم: (با فرض اینکه پیوندی)
 $O(E + E \log^* n) = O(E \log^* n)$
 $O(E \alpha(n))$