

روش کلی کار الگوریتم کروسکال به این شکل است، یک ساختمان داده جهت ذخیره یال ها و نود ها داریم تا گراف را به مدل کامپیوتری مدل سازی کنیم، حال باید یکی از این روش ها را برای انتقال گراف در حافظه را انتخاب کنیم و سپس یال ها را به ترتیب غیر نزولی (محدوده) مرتب کنیم و سپس هر بار یک یال را از لیست (مهم به زبان) انتخاب کنیم و تا جایی که دور تشکیل داده نشود به انتخاب یال ها ادامه بدهیم و همچنین اگر بیش از ۱-۱۹۹ تا یال انتخاب شده، الگوریتم را متوقف کنیم؛ درنت یال های انتخاب شده از نزولی به محدوده بدون ایبار دور MST ما خواهد بود.

الف) در روش ما ترتیب میاورتی نیاز داریم که برای ذخیره سازی گرافمان از یک ماتریس $V \times V$ برای مدل کردن گراف استفاده کنیم پس باید یال ها را مرتب سازی کنیم $E \log E$ (quick or merge sort) پس مرتبه زمانی این روش (ایبار ما ترتیب میاورتی + مرتب سازی) $O(E \log E + V^2)$

ب) در این روش از لیست استفاده می کنند، تعداد کل یال های (ذخیره شده) و برابر یال های گراف است و هزینه پیاده سازی این ساختمان داده $O(E)$ می باشد (در بدترین حالت $O(V^2)$)، در این جا هزینه اصلی مثل روش بالا از مرتب سازی یال ها گراف نیست می شود $O(E \log E)$ با استفاده از Quicksort یا mergesort است.

ج) در روش هیچ فیلد نیازی، هزینه افزودن یال ها $O(\log E)$ بود، هزینه ساخت اولیه هیچ (درخت ها و غیره) $O(1)$ بود و پس از ایبار کامل هیچ هزینه ای را از هیچ استخراج نمی کنیم $O(\log E)$ و چون E تا باید یال استخراج کنیم $O(E \log E)$ مرتبه زمانی ما خواهد بود.

تحریریات سر دوم [حریمان] استاد دکتر صالح پور

پارسیا یوسفی شاد. 1400536 11048

② ... به راه ملی پیشنهاد میدید.

3 راه ملی با دیدگاه های گوناگون را بررسی میکنیم:

● اگر بخواهیم راه یافت MST ، $PRIM$ یا کروسکال را ابراهیم کنیم تا MST بدست

بیاید، در صورتی که یال مورد نظر e در آن حضور داشته باشد، کار تمام است، در غیر این

صورت یال e با وزن M را به درخت اضافه ایجاد میکنیم، بدلیل اینکه درخت M را

یال داریم، افزودن یک یال افغانه e حتمی موجب ایجاد دور میشود، در دور بوجود آمده

بررسی میکنیم که آیا یال دیگری به وزن M داریم یا نه، اگر داشته باشیم، آن را حذف

میکنیم و بجایش یال دلخواه e ایمن را اضافه میکنیم، در صورتی که یالی با وزن مشابه یال

e یافت نشد، پس حتمی امکان حضور چنین یالی در MST ممکن نبوده چون حتمی دارای

وزن بیشترین نسبت به سایر یال های گراف داشته و توسط الگوریتم های گفته شده امکان

استثنا به هیچ وجه نداشته. اما اگر MST فراسته شده طرح لزومی MST بهینه (*) نداشته

و مد نظرش MST ای باشد که تحت هر شرایطی شامل یال e باشد، قضیه فرق میکند،

در این صورت میتوانیم بدون پس از افزودن یال e به مجموعه MST بهینه، در دور ایجاد شده

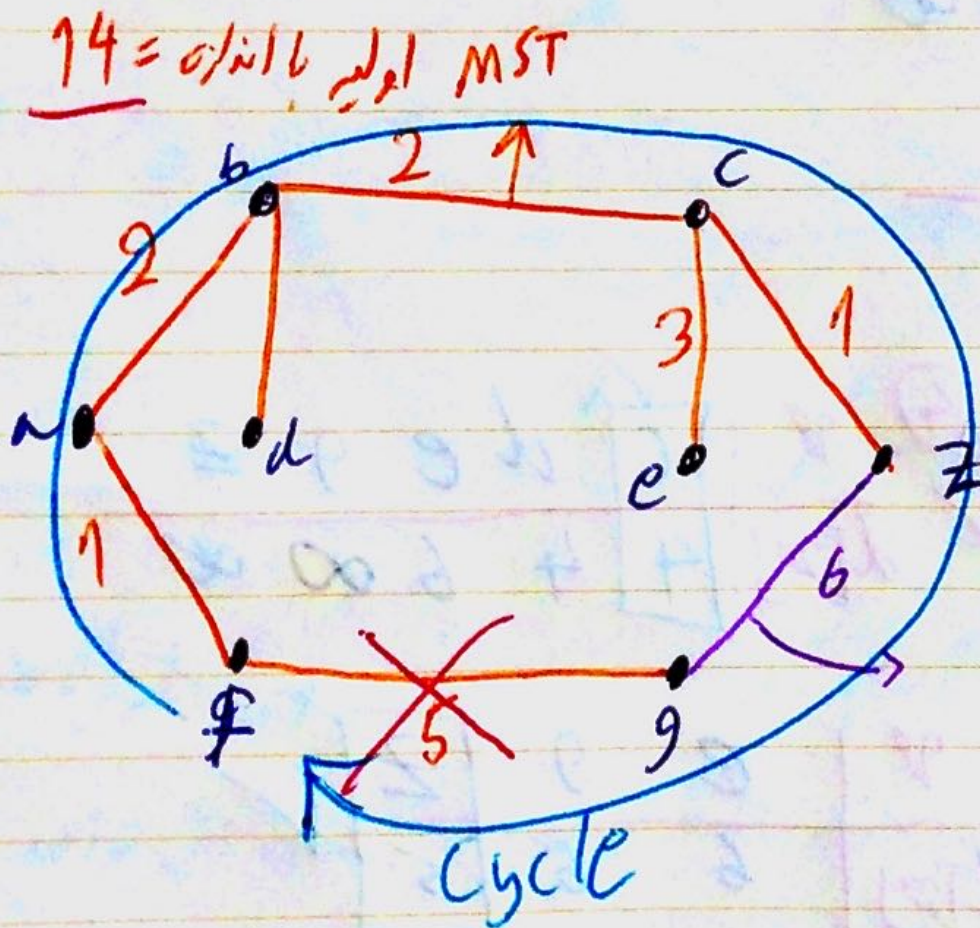
یال با بیشترین وزن را حذف و بجای آن e را با بیشترین کنیم تا به بهترین MST بدید

که حتمی یال e را در بر دارد دست پیدا کنیم.

یک راه حل دیگر این است که قبل از اجرای الگوریتم های کراسکال یا پریم، یال‌ها را از قبل انتخاب کنیم و سپس الگوریتم را اجرا کنیم. در این صورت یال‌ها حتماً در MST وجود خواهد داشت که لزوماً MST بهینه نبوده و پس حتماً شامل یال‌ها است.

مثال: با یک گراف سوال سوم؛ برای رد کردن دوم قطع

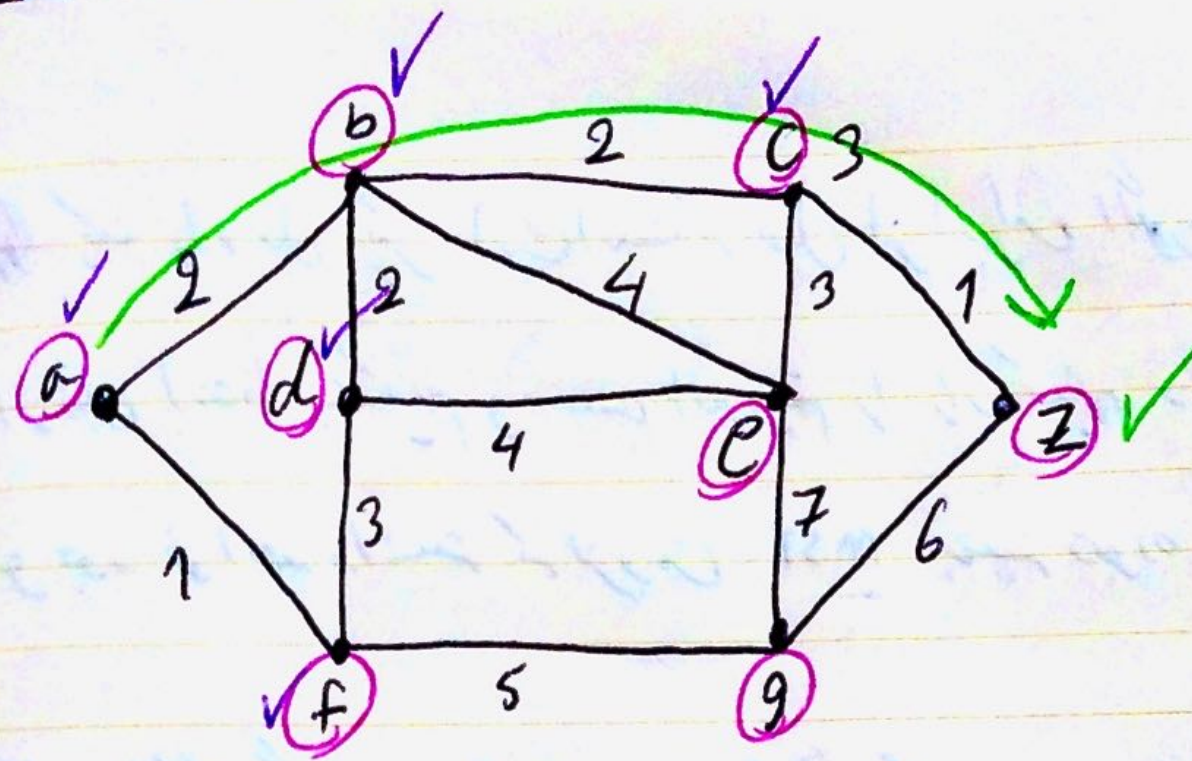
در نواحی یال e و حتماً موجود باشد در MST که پس آن MST جدید با MST واقعی فرق می‌کند:



با اندازه 15 $MST' = 6$ دور ایستاد
حذف یال f و
بزرگترین به 6

سوال

..... ذکر کنید.



توضیحات کامل و دقیق

①

u	a	b	c	d	e	f	g	z
$d[u]$	0	∞	∞	∞	∞	∞	∞	∞

update

②

u	b	c	d	e	f	g	z
$d[u]$	2	∞	∞	∞	1	∞	∞

③

u	b	c	d	e	g	z
$d[u]$	2	∞	4	∞	6	∞

④

u	c	d	e	g	z
$d[u]$	4	4	6	6	∞

⑤

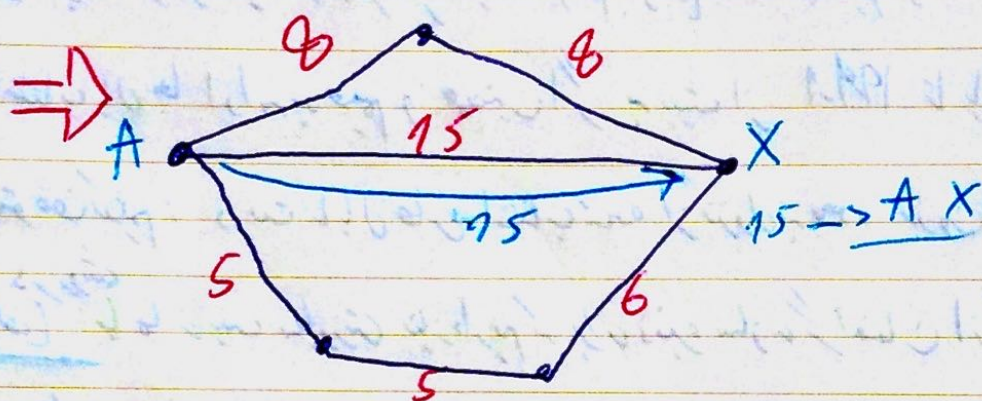
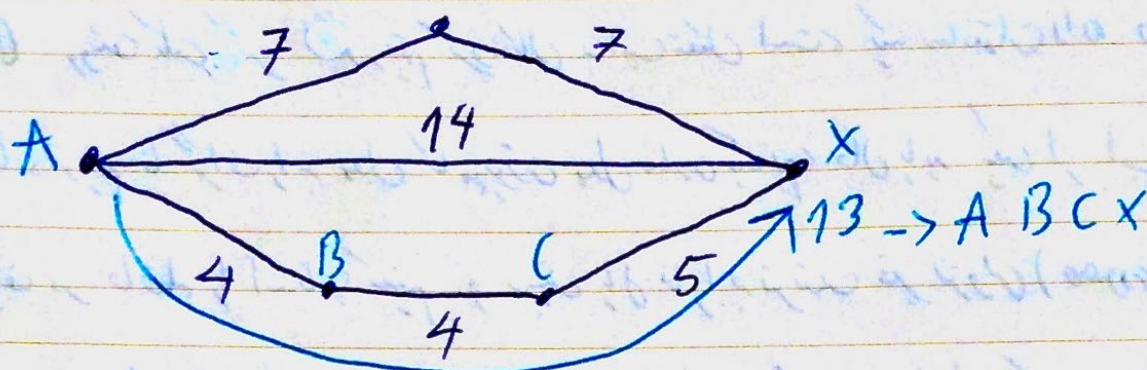
u	d	e	g	z
$d[u]$	4	6	6	5

⑥

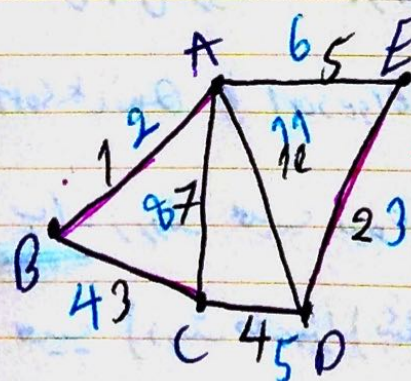
u	e	g	z
$d[u]$	6	6	5

← کمترین مسیری از a تا z = 5

4 ... اضافه کنید: الف) ... قواعد برد؟ مثال نقی طالب را مطرح کنیم:



ب) چگونه است؟ به ۱۱۵۶ برای مثال در الگوریتم کروسکال در هر مرحله به مرتب از یال ها
گراف ها اولاد اضافه کنیم و همچنین ترتیب مرتب شده یال ها از کم به زیاد مثل قبل قواعد
بود و الگوریتم طبق همان اولویت ها به ترتیب یال ها را از پایین به زیاد برین دارد و به همان
MST دست پیدا خواهیم کرد. فقط این وزن کل MST به اندازه تعداد یال ها نسبت به قبل



افزایش پیدا قواعد کرد: مثال:

edge	d_{29}	d_{29}^*
(A,B)	1	2
(E,D)	2	3
(B,C)	3	4
(C,D)	4	5
(A,E)	5	6
(A,)	7	8
(A,D)	10	11