

۹۸۴۱۱۴۳۲

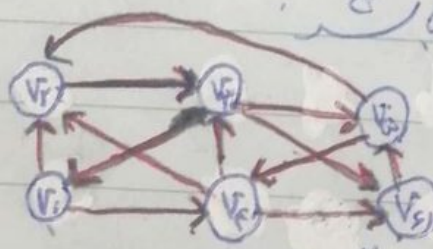
محمد عرفان زابع درسی

سن

با استفاده از BFS و عمق هر عنصر شروع به حل می کنیم. بدین گونه که یک مجموعه (آرایه یا لیست) را برای ذخیره عنصر کمینه در هر عمق در نظر می گیریم. بهیچان را از رأس خواسته شده (S) آغاز کرده و ابتدا متغیر عمق را به صفر در نظر می گیریم. سپس در هر مرحله (عمق) که با DFS در آن جلوی «برگ» متغیر را به عنوان «کمینه» داریم که در هر مرحله (قبل از ویرایش رتوس عمق جدید) بی نهایتی که از پیش. سپس عناصر را با متغیرمان مقایسه و در هر مرحله «کمینه» مقدار هر عمق را یافته و به مجموعه A می افزاییم. پس در هر مرحله زمانی که همه رتوس دیده شوند بحال با استفاده از جواب (اینده کسی که همان عمق است) به همراه مقدار «کمینه» را حساب می نماییم که صحت آن را جمع آوری می نماید. بخواهد که به دست آید کمینه در هر عمق است. کمینه جمع ممکن است به واسطه اساسی هر عمق (جواب) محاسبه شود.

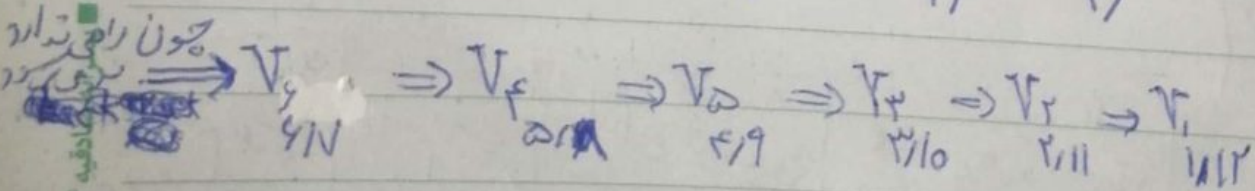
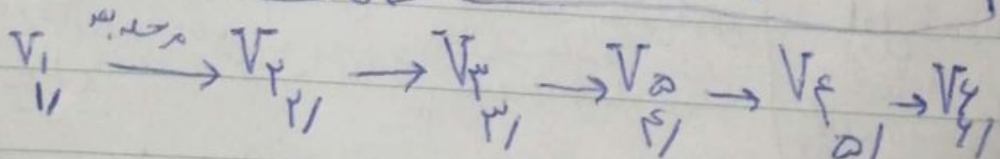
بهیچدی رضایی (الگوایم چون از BFS استفاده شده است از  $O(V+E)$  می باشد. و بهیچدی حافظه نیز با توجه به اینکه هر رأس بررسی و ذخیره شود و مجموعه A نیز تعدادی برابر عمق است، از  $O(V)$  می باشد.

$$4 \times 2 \times 2 = 9$$



د. س. ا. ف.   
 راهی نیست و برای هر   
 واحد بعد و پیشی در و گاهی   
 نکات

DFS:   
 شروع از  $V_1$



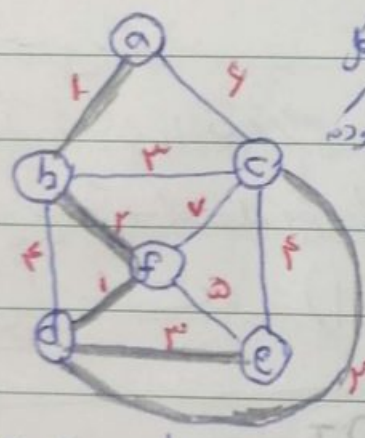


BFS:

شروع از رأس ۱:  $Q \rightarrow \boxed{V_1} \xrightarrow{V_1} Q \rightarrow \boxed{V_2 \ V_4} \xrightarrow{V_2} Q \rightarrow \boxed{V_4 \ V_3}$

$\xrightarrow{V_4} Q \rightarrow \boxed{V_3 \ V_6} \xrightarrow{V_3} Q \rightarrow \boxed{V_6 \ V_5} \xrightarrow{V_6} Q \rightarrow \boxed{V_5}$

$\xrightarrow{V_5} Q \rightarrow \emptyset$



با Kruskal سعی رویم. بدین گونه که یال ها را بر اساس وزن به شکل صعودی می تویم و هر بار که می خواهیم یالی اضافه شود بررسی می شود که آیا درخت A (مجموعه ی مورد نظر) در شکل نباید یعنی حلقه ایجاد کنند و عنصر دومی مشترک نباشند. حتم شروع می کنیم

~~ab~~ 1 ~~df~~ 1 ~~bf~~ 2 ~~cd~~ 2 ~~de~~ 3 ~~bc~~ 3 ~~ce~~ 4 ~~bd~~ 4 ~~fe~~ 5 ~~ac~~ 6

مرحله ۱:

اول طه ظاهر می شود یعنی اقترام چون یال کمینه است و حلقه ی قوی میخورد با نقص ندارد

مرحله ۲:

df اضافه می شود چون یال کمینه مانده است و حلقه ی برایش ساخت

مرحله ۳:

bf هم به همان درای قبل افزوده شود

مرحله ۴:

cd نیز شرط برقراره و کمینه یال مانده است

مرحله ۵:

de نیز ادویه و شرط برقراره کمینه مانده است

مرحله ۶:

bc ولی هر دو عنصر b و c در رأس ها و حلقه ی دو رأسی می شود (سی در نظر نمی گیریم)

مرحله ۷:

ce نیز هر دو رأسی آن در A اند و ایجاد حلقه می کنند سه در نظر نمی گیریم

مرحله ۸:

bd نیز حلقه ی سی در نظر نمی گیریم

مرحله ۹:

fe هم حلقه ی سی در نظر نمی گیریم (مجموعه ی دو رأسی آن A است)

مرحله ۱۰:

ac هم حلقه ی سی در نظر گرفته نمی شود



همان طور که مشاهده می شود با اجرای الگوریتم Kruskal روی گراف به نتایج بالا رسیدیم که یال های ناموافق حذف گردیدند و در نظر ما هستند یال های که با قریز خط خوردند یعنی قطع نشده شرط بودند.

$$1+1+2+2+3=9$$

سایر موارد در نظر هم شد:  
پیمودگی زمانی الگوریتم از مرتبه  $O(E \log V)$  می باشد. پیچیدگی حافظه ای هم به اندازه تعداد رئوس به علاوه یال های ناموافق است  $O(E+V)$ .

## سؤال ۴

دو درخت به نام های  $A_1$  و  $A_2$  را در نظر می گیریم. این دو درخت از حذف عدن یال  $(u, v)$  حاصل می شوند که از MST حذف شده.

حال ۵ رأس در هر کدام از  $A_1$  و  $A_2$  به نام های  $v_1$  و  $v_2$  در نظر می گیریم حال

$cut$  جدا کننده این دو  $(v_1, v_2)$  بررسی می کنیم و در نظر می گیریم که اگر عرض نمی

در  $cut$  ایجاد شده یالی موجود است که وزن آن از یال  $(u, v)$  کمتر و یا برابر

آن گاه می شود به حاصل اجتماع دو درخت اولیه که از حذف  $(u, v)$  حاصل

شده اند افزودن یک درخت به توانا کنیم داشته  $(A_1, A_2)$  که این

سبب ایجاد MST با وزن کمتر از MST اولیه می شود که دارای  $(u, v)$  بود

و نشان می دهد MST اولیه نادرست است. که این خلاف قضیه است

پس با توجه به این ۶ چیز می توان گفت و با توجه به تعریف  $Light$  (یالی که بین

دو  $cut$  را به هم وصل و کمترین وزن را دارد داخل MST است) را

دست ایما می دهیم

$(u, v)$  قطعه ای که  $Light$  می باشد  
edge