دانشکده‌ی علوم ریاضی

آنالیز الگوریتم ها	مدرس: دکتر شهرام خزایی

تمرین سری دو

نام و نام‌خانوادگی : کژال باغستانی — *شماره دانشجویی:* ۴۰۱۱۰۰۰۷۱

پرسش ۱

برای حل این مسأله هر بار یکی از رأس های گراف را انتخاب میکنیم و یال های متصل به آن را حذف میکنیم سپس BFS را اجرا میکنیم و مولفه های همبندی گراف باقیمانده را پیدا میکنیم و درنهایت به تعداد مولفه های همبندی معادله داریم به صورتی که مثلا اگر x_1 و x_2 و x_3 در یک مولفه و x_4و x_5 در یک مولفه و x_6 نیز به تنهایی در یک مولفه باشند معادله ای که بدست می آوریم به صورت زیر است:

$$x_1 + x_2 + x_3 = x_4 + x_5 = x_6$$

که معادله های زیر را به ما میدهد

$$x_1 + x_2 + x_3 - x_4 - x_5 = 0$$

$$x_4 + x_5 - x_6 = 0$$

$$x_1 + x_2 + x_3 - x_6 = 0$$

برای تمام این معادله های به دست آمده یک ماتریس ضرایب $n \times n$ درست میکنیم و نام آن را A میگذاریم حالا n رأس را به صورت $x = (x_1, x_2, .., x_n)^T$ نامگذاری میکنیم و معادله $Ax = 0$ را حل میکنیم که حل این معادله با عملیات سطری پلکانی $O(n^3)$ زمان میبرد و هر بار BFS روی هر یک از رأس ها برای پیدا کردن مولفه های همبندی نیز $O(n)$ پس در مجموع $O(n^2)$ زمان میبرد که در کل الگوریتم در زمان چند جمله ای اجرا میشود. توجه کنید که n معادله n مجهول ممکن است چند متغیر آزاد به ما بدهد که میتوانیم وزن آنها را به دلخواه انتخاب کنیم.

پرسش ۲

با گرفتن لیست اسکرین شات نفر اول ترتیب تمام افراد به جز نفر اول را داریم گراف جهت داری از $O(n)$ میسازیم به صورتی که نفر i به نفر j به صورت جهت دار متصل میشود اگر نفر i در لیست فعالیت ها بالا تر باشد. حالا از روی لیست نفر دوم اسم نفر اول را پیدا کرده و بین دو نفری که قرار دارد یال های جهت دار را رسم میکنیم در بدترین حالت دو راس نفر اول و نفر i ام هر دو بین دو نفر j و k قرار میگیرند که در اسکرین شات سوم ترتیب نفر اول و i نیز مشخص میشود. در کل در بدترین حالت با سه اسکرین شات لیست افراد تشکیل میشود.(ممکن است با دو اسکرین شات نیز این لیست تشکیل شود) و دوباره از روی تمام اسکرین شات ها این ترتیب را چک میکنیم که دقیقا همان گرافی که ساختیم با غیبت یک راس که خود شخص فرستنده است باشد. و اگر گراف به صورت

یکتا مشخص شده بود و اسکرین شات ها تناقضی نداشتند به سادگی با پیمایش روی گراف ساخته شده اسامی را خروجی میدهیم و در غیر این صورت میگوییم هیچ ترتیب زمانی وجود ندارد. زمان این الگوریتم به وضوح $O(kn)$ است.(k اسکرین شات داریم و پیمایش و چک کردن هر کدام $O(n)$ زمان میبرد).

پرسش ۳

ابتدا از روی لیست f_i ها گرافی جهت دار که راس های آن کارمندان و یال های آن به صورت جهت دار از هر کارمند به کارمندی است که باید به آن خبر رسانی کند تشکیل میدهیم. سپس مولفه های قویا همبند را $O(|V|+|E|)$ پیدا میکنیم که در اینجا تعداد یال ها دقیقا با تعداد راس ها برابر است (زیرا هر کارمندی دقیقا به یک کارمند یال جهت دار دارد.) پس $O(n)$.

ادعا:

هر شخصی خبری که میشنود به همه افراد منتقل میشود اگر و تنها اگر تمام افراد در یک مولفه قویا همبند قرار گیرند.

اثبات :

اگر تمام افراد در یک مولفه قویا همبند قرار گیرند به وضوح از هر راس (کارمند) به راس های دیگر مسیری جهت دار هست پس خبر به خوبی پخش میشود.

حال میخواهم نشان دهم اگر تمام افراد در یک مولفه قویا همبند نباشند فردی وجود دارد که اگر خبر به او برسد همه مطلع نمیشوند. می دانیم هر dag حداقل یک sink دارد پس اگر خبر به یکی از افراد در مولفه ای که sink است برسد خبر فقط در همان مولفه میماند و افراد در مولفه های دیگر از آن خبر مطلع نمیشوند پس ادعا ثابت میشود.

ادعا:

کمترین یال مورد نیاز برای همبند کردن گراف dag برابر است با $max(a,b)$ که a تعداد رأس ها با درجه ورودی صفر و b تعداد رأس ها با درجه خروجی صفر است.

اثبات:

اگر از هر رأس به رأس دیگر راه وجود داشته باشد حتما تمامی رأس ها حداقل یک یال ورودی و خروجی دارند. پس حداقل به تعداد $max(a,b)$ نیاز داریم. حالا الگوریتمی ارائه میدهم که با این تعداد dag را تبدیل به گراف قویا همبند کند. توجه کنید که راس هایی که یا درجه ورودی ندارند یا درجه خروجی یا سینک هستند و یا سورس . ابتدا مشخص میکنیم که a بیشتر است یا b فرض کنیم که a>b داریم:

سورس ها بیشتر از سینک ها در نتیجه هر سینک را به سورس بعدی آن وصل میکنیم برای ایجاد دور همیلتونی و اگر دو یا چند سورس بعد از سینک مورد نظر بود سینک مورد نظر را به هر دو وصل میکنیم به صورت شهودی مانند قطعه های پازل گراف های dag را میچینیم و سینک های قطعه قبلی را به سورس های قطعه بعدی متصل میکنیم. با اینکار به وضوح دور همیلتونی تشکیل میشود و تعداد یال های مورد نیاز به تعداد سورس هاست. اگر سینک ها بیشتر بود یعنی b>a نیز به صورت مشابه عمل میکنیم و الگوریتمی ارائه دادیم که با $max(a,b)$ دور همیلتونی ایجاد میکند و اثبات کامل میشود.

پرسش ۴

پرسش ۵

ابتدا الگوریتم DSF را روی ریشه درخت اجرا کرده به این صورت که به هر رأس دو مقدار pre و post را نسبت میدهیم که اگر یک counter داشته باشیم که نشان دهنده زمان است مقدار pre هر رأس زمانی است که آن رأس ملاقات شده و مقدار post آن رأس زمانی است که می خواهیم آن رأس را ترک کرده و دیگر فرزندی برای پیمایش ندارد. پس تا اینجا درخت DFS را برای رأس ها تشکیل میدهیم و به هر رأس مقدار $[pre,post]$ را نسبت میدهیم.

که همان اردر زمانی DFS را دارد.$O(n)$.
حالا به بررسی query داده شده میپردازیم.
فرض کنیم در پرسش داده شده رأس های $v_1, ..., v_i$ به ما داده میشود. برای اینکه نشان دهیم آیا مسیری از ریشه وجود دارد که شامل همه اینها باشد یا حداقل به فاصله یک از اینها باشد چند ادعا را ثابت میکنم.
ادعا۱:
اگر در میان v_j ها رأس با بیشترین مقدار pre را در نظر بگیریم مسیر این رأس به ریشه همان مسیر مورد نظر است.
اثبات:
رأس با بیشترین مقدار pre رأسی هست که دیر تر از همه رأس ها ملاقات شده پس اگر بخواهیم مسیر ما شامل این رأس باشد باید از خودش بگذرد زیرا مسیر رأس های دیگر به ریشه هیچ وقت به این رأس نمیرسد چون هر گاه به ریشه میرویم یا به عبارت دیگر به سمت بالا میرویم به رأس هایی به مقادیر pre کمتر میرسیم چون آنها زودتر ملاقات شدند که به ریشه نزدیک ترند. درنتیجه اگر چنین رأسی وجود داشته باشد حتما همان رأس با بیشترین مقدار pre است.
ادعا ۲:
رأس b در زیر درخت رأس a است اگر و تنها اگر $pre(a) < pre(b)$ و $post(a) > post(b)$.
اثبات:
اگر $pre(a) < pre(b)$ باشد یعنی رأس b دیر تر ملاقات شده و از طرف دیگر اگر $post(a) > post()$ باشد یعنی ابتدا از b برگشتیم و سپس به a رسیدیم پس a جز اجداد b است.
و اما طرف دیگر اگر a جز اجداد b باشد اول a ملاقات میشود و سپس b و بعد از b برمیگردیم به a و در نهایت از a برمیگردیم پس داریم:$pre(a) < pre(b)$ و $post(a) > post(b)$. که ادعا ثابت میشود.
ادعا ۳:
در مسیر مورد نظر ما بالاترین رأس در درخت DFS بین v_j ها رأسی است که کمترین مقدار pre را دارد.
اثبات :
به وضوح مانند ادعا۱ ثابت میشود.
ادعا۴:
برای اینکه ببینیم رأس v_j در این مسیر هست یا آیا فاصله یک با این مسیر دارد کافیست بررسی کنیم که آیا خودش در این مسیر هست یا پدرش یا خیر.
اثبات:
به وضوح اگر خودش در این مسیر باشد ادعا ثابت میشود. حال میخواهم نشان دهم اگر بخواهیم یک فاصله با این مسیر داشته باشد کافیست پدرش در این مسیر باشد. رأس هایی که در مسیر هستند فقط به پدرشان و فرزندانشان متصل هستند زیرا درخت DFS یال بازگشتی ندارد و از طرف دیگر چون گراف ورودی درخت است cross edge نیز نداریم. از طرف دیگر چون گراف ورودی درخت است هر رأس فقط یک پدر دارد در غیر این صورت دور تشکیل میشود. از طرف دیگر اگر فرزند یک رأسی در یک مسیر به ریشه باشد آن رأس نیز به علت اینکه هر رأس یک پدر دارد در آن مسیر هست. پس تنها حالتی که یک رأس از یک مسیر فاصله یک دارد این است که پدرش در آن مسیر باشد.
ادعا ۵:
یک رأس در یک مسیر بین دو رأس a و b هست اگر و تنها اگر آن رأس در زیر درخت های a باشد و b در زیر درخت های آن رأس باشد. (با فرض اینکه ارتفاع b بیشتر از ارتفاع a است.)
اثبات:
از آنجایی که مسیر بین a و b در درخت یکتاست پس به وضوح این ادعا برقرار است.

حالا با کمک ادعا هایی که ثابت کردم برای پرسش i ام کافیست دو رأس a و b را با گرفتن min و max

روی مقادیر pre رأس ها مشخص کنم.($O(k_i)$) و در نهایت برای هر رأس بررسی کنم که آن رأس یا پدرش در زیر درخت a باشند$O(1)$ و b جز رأس های زیر درخت آن رأس یا پدرش باشد.$O(1)$. که تعداد این رأس ها برابر با k_i هست و از طرف دیگر تعداد کل پرسش ها m تاست و $\sum k_i = k$ پس زمان کل الگوریتم برابر است با: $O(n + m + k)$.

پرسش ۶

برای اثبات این الگوریتم کافیست دو چیز را ثابت کنیم که به صورت دو ادعا در زیر ارائه میکنم.

در ابتدا به این نکته توجه میکنیم که در درخت هر گاه مسیری بین دو رأس داریم آن مسیر تنها مسیر بین آن دو رأس است. (زیرا در غیر این صورت دور تشکیل میشود) درنتیجه مسیری که بین دو رأس i و j در درخت BFS وجود دارد ابتدا از i به ریشه میرویم و اگر j در این مسیر نبود از j به سمت ریشه میرویم و اولین جایی که به مسیر از i به ریشه برخورد کردیم میتوانیم به سمت i برویم. پس هر مسیری یک بخش بالا رونده به سمت ریشه و یک بخش پایین رونده به سمت رأس مورد نظر دارد. و نکته دوم قابل توجه این است که قطر درخت هرگز نمیتواند بین i و j ای باشد که یکی در مسیر دیگری به ریشه است. زیرا فاصله آن رأس تا ریشه بیشتر از فاصله آن تا رأس دیگر میشود.

ادعا ۱:
اگر قطر گراف طول D داشته باشد هر رأس را به عنوان ریشه درخت BFS انتخاب کنیم حتما یک سر قطر در سطح آخر می افتد.
اثبات:
رأس v را به عنوان ریشه درخت BFS در نظر میگیریم. سر اول قطر از آن فاصله H و سر دوم قطر از آن فاصله $D - H$ دارد. در این صورت ارتفاع درخت BFS برابر با $max(H, D - H)$ میشود. زیرا اگر بیشتر شود رأسی به فاصله بیشتر از این دو، از ریشه درخت وجود دارد که این یعنی آن رأس به عنوان سر دیگر قطر شناخته میشود.

ادعا ۲:
اگر در درخت BFS یک راس از سطح آخر انتخاب کنیم و BFS را روی آن اجرا کنیم مستقل از اینکه آن کدام رأس است عدد یکسانی به عنوان قطر به ما میدهد.
اثبات:
فرض کنیم $a_1, a_2, .., a_n$ رأس های سطح آخر ما هستند. اگر a_i راس i را به عنوان سر دیگر قطر خروجی دهد اگر a_j را نیز در نظر بگیریم . یا جد مشترک آن با i همان جد مشترک a_i با i است که در این صورت همان فاصله را با i دارد. و یا جد مشترک آن با i متفاوت از جد مشترک a_i و i است که در این صورت اگر در نظر بگیریم جد مشترک a_j و i پایین تر از جد مشترک a_i و i است. اگر i در سطح آخر نباشد آنگاه از جد مشترک i و a_j بجای رفتن به i به a_j میرویم. که این یعنی طول قطر برای a_i بیشتر میشد. پس اگر جد مشترک متفاوت داشتیم نیز فاصله a_j و a_i با هم را به عنوان خروجی برای a_j در نظر میگیریم. (زیرا برابر با فاصله a_i و i بود.) حالا کافیست نشان دهیم که هیچ کدام از a_i ها نمیتوانند عدد بزرگ تری از دیگری به قطر نسبت بدهند. این بخش نیز به وضوح معلوم است چون نشان دادیم هر عددی که از هر کدام از a_i ها بگیریم به سادگی میتوان به a_j دیگر نیز راسی نسبت داد که همان عدد را بدهد. پس همگی مساوی و ماکسیمم نداریم.

و اثبات کامل میشود.