به نام خدا

ساختمان داده‌ها و الگوریتم‌ها (۴۰۲۵۴)
دانشگاه صنعتی شریف
مدرس: دکتر مهدی صفرنژاد

تمرین پنجم
انتشار: ۵ دی ۱۴۰۰

گراف

سؤالات را با دقت بخوانید و روی همه آن‌ها وقت بگذارید. تمرین‌های تئوری تحویل گرفته نمی‌شوند اما از آن‌ها سؤالات کوییز مشخص می‌شود. بنابراین روی سؤالات به خوبی فکر کنید و در کلاس‌های حل تمرین مربوطه شرکت کنید.

سؤال ۱. تورنمنت یک گراف کامل جهت‌دار است. به دو سوال زیر پاسخ دهید.

۱. ثابت کنید در هر تورنمت راسی وجود دارد که عمق درخت BFS مربوط به آن راس حداکثر ۲ است.

۲. باتوجه به قسمت قبل ثابت کنید اگر هیچکدام از رئوس درجه‌ی خروجیشان $n-1$ نباشد، حداقل دو راس با چنین خاصیتی داریم.

پاسخ:

۱. دقت کنید حکم داده شده در قسمت اول معادل با این است که ثابت کنیم در یک تورنمت راسی وجود دارد که با حداکثر دو یال به تمامی رئوس مسیر دارد.(چنین راسی را شاه می نامیم)برای اثبات از استقرا استفاده می کنیم.پایه استقرا $n = 1$ و $n = 2$ است که حکم برای این دو پایه بدیهی است.برای گام استقرا ابتدا راس دلخواه a را حذف کنید.روشن است که یک تورنمت با $n - 1$ راس باقی می ماند.طبق فرض استقرا این تورنمت شاهی دارد که آن را b می نامیم.حال راس a را به گراف بازگردانید.اگر b به a یال داشته باشد که کار تمام است و b همچنان شاه است.نتیجتا فرض می کنیم a به b یال دارد.حال رئوسی را که b با آن ها یال دارد را درنظر بگیرید.اگر هریک از این رئوس به a یال داشته باشند آنگاه b با دو یال به a مسیر دارد که این یعنی مجددا b شاه می ماند.فرض کنید a به تمامی این رئوس یال داشته باشد.با کمی دقت می توان فهمید که در این صورت a از نظر سطح دسترسی مانند b شده است(تمامی رئوسی که b به آنها یال داشت a هم یال دارد پس تمامی رئوسی که b با دو یال به آنها مسیر داشت a هم با دو یال به آنها مسیر دارد) با این تفاوت که a به خود b هم یال دارد نتیجتا a شاه جدید تورنمت است.

۲. چون راسی نیست که درجه خروجی اش $n - 1$ باشد نتیجتا سطح دوم درخت BFS خالی نیست.از طرفی می دانیم که تورنمت قطعا یک شاه را دارد. این شاه و تمام رئوسی که به آن ها یال دارد را حذف کنید. گراف باقی مانده مجددا یک تورنمت است و شاه مخصوص به خودش را دارد.همچنین بدیهی است که شاه این تورنمت کوچکتر باید به شاه تورنمت بزرگتر یال داشته باشد(چرا که در عمق دوم درخت bfs شاه اول قرار داشت) پس شاه تورنمت کوچکتر به تمام رئوسی که شاه اول با یک یال به آنها مسیر دارد با دو یال مسیر دارد.از تمامی این جملات جمع بندی می‌کنیم که این شاه دوم هم یک شاه برای کل تورنمت است.

سؤال ۲. فرض کنید که G یک گراف همبند و بدون جهت است که یکی از راس‌هایش s است. اگر BFS و DFS را از این راس شروع کنیم و هر دو درخت حاصل الگوریتم‌ها یکسان باشند، ثابت کنید که گراف G یک درخت است.

پاسخ:

فرض کنیم که درختی که توسط DFS و BFS تولید می‌شود T باشد. ابتدا در نظر بگیرید که G یک درخت نیست. این یعنی یک یال (u,v) در G است که در T نیست.در این حالت در درخت DFS یکی از u یا v باید پدر دیگری باشند. در درخت BFS هم u و v باید یک سطح با هم فاصله داشته باشند. چون هر دو درخت یکسان هستند، پس یکی از u یا v پدر دیگری است و یک سطح هم باید با هم فاصله داشته باشند. بنابراین این یال حتما باید در T باشد که با فرض ما مخالف است. بنابراین G حتما یک درخت باید باشد.

سؤال ۳. یک الگوریتم برای تشخیص اینکه آیا یک گراف بدون جهت، حاوی یک دور است یا خیر ارائه دهید. در درصورت وجود، الگوریتم شما باید یک دور را خروجی دهد و همچنین زمان اجرای الگوریتم شما باید برای گرافی با n گره و m یال $O(m+n)$ باشد. پاسخ:

روی گراف ورودی DFS می‌زنیم و درخت حاصل آن را T می‌نامیم. اگر $T=G$، آنگاه هیچ دوری نداریم اما در غیر این صورت یال (u,v) را که در G هست ولی در T نیست، پیدا می‌کنیم. مسیر u به v به نام P را پیدا می‌کنیم و این مسیر و خود یال (u,v) را که یک دور تشکیل می‌دهند خروجی می‌دهیم.

سؤال ۴. گراف جهت دار و بدون دور G را در نظر بگیرید. الگوریتمی از مرتبه $O(m+n)$ ارائه دهید که بلند ترین مسیر گراف را پیدا کند.(m تعداد یال ها و n تعداد راس های گراف است.)

پاسخ: مسئله را برای حالت کلی تر بررسی میکنیم. فرض کنید گراف جهت دار وزن دار و بدون دور G با وزن های مثبت داریم. ابتدا در زمان $O(m+n)$، $topological-order$ راس های گراف را بدست می آوریم.سپس یک آرایه $dist$ به طول n تعریف میکنیم و در ابتدا همه ی مقادیر آن را منفی بینهایت قرار میدهیم. سپس به ترتیب $topological-order$ جلو میرویم و در هر مرحله به ازای هر راس u عملیات های زیر را انجام می‌دهیم:

به ازای همه ی همسابه های u مانند v اگر $dist[u]+weight(uv)>dist[v]$ آنگاه قرار می‌دهیم:

$dist[v]=dist[u]+weight(uv)$. در واقع از آن جایی که هر مسیری که در گراف داشته باشیم روی $topological-sort$ مسیری رو به جلو است، با این الگوریتم همه ی مسیر های ممکن را بررسی میکنیم. در آخر $max(dist)$ جواب مسئله خواهد بود. سورت ابتدایی در زمان $O(m+n)$ انجام میشود سپس در ادامه ی الگوریتم هر یال دقیقا یکبار بررسی میشود که مرتبه زمانی $O(m)$ دارد و در نهایت نیز ماکسیمم عدد های داخل آرایه را بدست می آوریم که از مرتبه $O(n)$ است. در نتیجه کل الگوریتم در مرتبه زمانی $O(m+n)$ انجام میشود.

سؤال ۵. فراز یک گراف جهت دار n راسی به عنوان کادوی تولد ۲۰ سالگی‌ش دریافت کرده است. پس از کمی بررسی، فراز متوجه شد که هر راس دقیقا یک خروجی دارد. محمد که نمی‌خواهد کادوی تولد او خسته کننده باشد، از فراز می‌پرسد که طول تمام دورهایی که در این گراف وجود دارد را در $O(n)$ بیابد.

پاسخ:

با کمی بررسی به این نتیجه می رسیم که گراف از چند دور و مسیر که به این دور ها می رسد تشکیل شده است. با اجرای الگوریتم dfs از یک راس به یک دور می رسیم. برای هر راس سه حالت در نظر می گیریم اول اینکه تا به حال از آن dfs نزده ایم دوم اینکه

آن را تمام کرده ایم و سومین حالت اینکه dfs آن را شروع کرده ایم ولی هنوز تمام نکرده ایم. یک دور در حالتی شکل می گیرد که به یک راس که در حالت سوم است برویم. و اگر در حالت اول بود کار را ادامه می دهیم و اگر در حالت دوم بود به آن وارد نمی شویم. حال پس از پیدا کردن یک راس که در دور است از آن راس dfs می زنیم و تعداد راس هایی که می بینیم تا به همان راس برگردیم را در یک متغیر ذخیره می کنیم. سپس از ما ماکسیمم این اعداد خواسته شده است که می توانیم به راحتی با ذخیره ماکسیمم و آپدیت مقدار آن پس از هر مرحله آن را به دست آوریم.

سؤال ۶. فراز پس از مواجهه با سوال قبلی، ارتباط خوبی با دورها گرفته است. او به صورت اتفاقی یک تورنومنت (گراف کامل جهت‌دار) یافت و مشاهده کرد که تعداد زیادی دور در این تورنمنت وجود دارد. حال او می‌خواهد بداند آیا می‌توان در یک تورنمنت n راسی، در اردر $O(n^2)$ یک دور ۳ تایی (در صورت وجود) یافت؟

پاسخ:

در هر تورنومنت اگر مثلث جهت‌دار نداشته باشیم آنگاه تعداد کل مثلث های غیر جهت‌دار برابر انتخاب ۳ از n است. از طرفی هر مثلث غیر جهت دار دقیقا یک راس دارد که یکی از یال های مثلث به آن وارد و دیگری از آن خارج میشود که آن را نماینده مثلث می‌نامیم. در این صورت اگر برای شماردن مثلث های غیر جهتدار تعداد رئوس نماینده را بشماریم، به ازای هر راس با درجه d، این راس نماینده دقیقا $d(n-d-1)$ مثلث خواهد بود. پس اگر به ازای تمام رئوس، مقدار گفته شده را جمع بزنیم برابر تعداد کل مثلث ها خواهد بود. در این صورت برای فهمیدن وجود دور به طول ۳، کافی است این دوگانه شماری را چک کنیم چون تنها زمانی درست است که مثلث جهتدار نداشته باشیم.

سؤال ۷. فراز که در گراف اوستا شده است، آوازه‌ش به عنوان یک گراف‌شناس معروف در جهان به گوشه همه رسیده است. به همین دلیل، از او دعوت شده است که در شهری عجیب، یک سمینار علمی برگزار کند. اما بعضی از دانشمندان این شهر در توانایی‌های فراز تردیدهایی دارند. در این شهر، n میدان وجود دارد و بین برخی از میدان‌ها، جاده‌هایی وجود دارد. می‌توان به میدان‌ها به شکل رئوس یک گراف نگاه کرد و جاده‌ها، یال‌هایی بین رئوس هستند. در این شهر، بین هر دو میدانی، دقیقا یک مسیر وجود دارد. ضمنا جاده‌ها، طول‌های متفاوتی می‌توانند داشته باشند (طول‌ها اعدادی صحیح و مثبت هستند). حال دانشمندان شهر، نقشه‌ی کامل شهر را به همراه k جفت از میدان‌ها به فراز داده‌اند. فراز باید در اردر $O(n+k)$ k عدد به آن‌ها بدهد که عدد i ام، فاصله‌ی دو میدانی است که در جفت i ام ذکر شده‌اند. در ضمن فاصله‌ی دو میدان، برابر با XOR اعداد مسیر ساده‌ی بین دو میدان است. آیا می‌توانید که الگوریتمی مناسب برای فراز طراحی کنید و او را از این چالش به سلامت عبور دهید؟

پاسخ:

چون بین هر دو راس دقیقا یک مسیر وجود دارد پس گراف ما درخت است. در این صورت کافی است یک راس دلخواه را به عنوان ریشه درخت انتخاب کنیم و از آن راس الگوریتم dfs را انجام دهیم به طوری که آرایه ای به نام cost به طول n در نظر میگیریم. مقدار cost برای راس ریشه را ۰ در نظر میگیریم و در پیمایش در dfs مقدار راس فرزند را برابر حاصل xor مقدار cost پدر با یال بین آن ها در نظر میگیریم. پس از این عملیات برای هر جفت راس کافی است cost دو راس را xor کنیم و به حاصل مورد نظر دست یابیم.

سؤال ۸. یک جدول $m \times n$ داریم که در هر خانه از آن یا یک انسان قرار دارد یا یک زامبی. هر روز که میگذرد، هر زامبی همسایه هایش را (همسایه های ضلعی) اگر انسان باشند تبدیل به زامبی میکند و این روند ادامه پیدا میکند. الگوریتمی از مرتبه $O(mn)$ ارائه دهید که تعداد روز هایی که طول میکشد که کل جدول زامبی شوند را بدست آورد.

پاسخ: هر کدام از خانه های جدول را یک راس از گراف در نظر میگیریم. در این صورت درجه ی هر راس برابر با تعداد خانه های مجاور آن میشود. یک راس جدید به نام x میسازیم و x را به همه ی راس هایی که در ابتدا زامبی هستند وصل میکنیم. سپس از x BFS میزنیم. ارتفاع درخت بدست آمده، جواب مسئله است. چرا که اگر یک انسان در عمق k از این BFS قرار داشته باشد، یعنی اینکه نزدیک ترین زامبی به این انسان فاصله اش k است در نتیجه حداقل k روز طول میکشد تا این انسان تبدیل به زامبی شود. تعداد راس های گراف mn و تعداد یال ها $\approx 4mn$ است. در نتیجه مرتبه زمانی این BFS، $O(mn)$ است.

سؤال ۹. برای گرافی وزن دار، بدون جهت، ساده و همبند G با n راس و m یال، عدد b را گلوگاهی می‌گوییم، اگر b بزرگ ترین عددی باشد که بین هر دو راس، مسیری وجود داشته باشد که همه ی یال های آن مسیر حداقل b باشد. الگوریتمی با زمان اجرای $O(mlogn)$ برای محاسبه عدد گلوگاهی ارائه دهید.

پاسخ: یال ها را بر اساس وزن آن ها مرتب می‌کنیم و در یک آرایه می‌ریزیم. حال با استفاده از جستجوی دودویی عدد گلوگاهی را پیدا می‌کنیم. در هر مرحله عنصر وسط(که آن را x می‌نامیم) را در نظر می‌گیریم و یال هایی که وزنشان بزرگ تر یا مساوی x است را در نظر می‌گیریم. اگر گراف حاصل همبند باشد، باید داشته باشیم: b ≤ x

تا به اینجای کار روی یال‌ها جستجوی دودویی انجام دادیم که خود هزینه‌ای معادل $O(\log m)$ دارد، اما توجه داشته باشید که تعداد یال ها حداکثر برابر $\binom{n}{2}$ هستند که از این موضوع می‌توان نتیجه گرفت هزینه نهایی هر مرحله برابر $O(\log n) = O(\log n^2)$ است. حال در هر مرحله پس از اجرای جستجوی دودویی باید همبند بودن گراف حاصل را بررسی نماییم. این کار با استفاده از جستجوی عمق نخست (dfs) که هزینه ای معادل $O(n+m)$ دارد امکان‌پذیر می‌باشد، همچنین مانند حالت قبلی با استدلال مشابه کوچکتر بودن اندازه مجموعه رئوس از یال‌ها می‌توان نشان داد این مقدار از $O(m)$ می‌باشد. بنابر این زمان کلی الگوریتم از $O(m \log n)$ خواهد بود.

موفق باشید