

ساختمان داده‌ها و الگوریتم‌ها

پاسخ تمرین پنجم - گراف

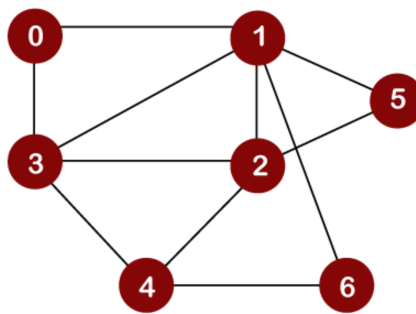
مجید فریدفر

تاریخ تحویل: ۱۴۰۳/۳/۲۴

۱۵ نمره

۱. BFS و DFS

الگوریتم‌های dfs و bfs را روی گراف زیر اجرا کنید. راس صفر را ریشه در نظر بگیرید و فرض کنید هر جا که قابلیت انتخاب بین چند راس وجود داشت، راسی که مقدار آن کم‌تر است اولویت بالاتری دارد.

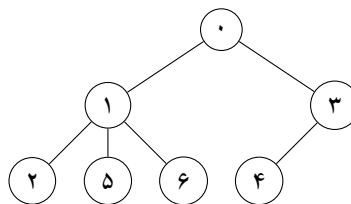


- الف) زمان ورود و خروج (starting time و finishing time) را برای هر راس مشخص کنید.
ب) مشخص کنید یال‌های خارج از درخت dfs و bfs از چه نوعی‌اند (cross-edge یا back-edge).
ج) درخت حاصل از اجرای این الگوریتم‌ها را رسم کنید.

پاسخ:

BFS:

Tree:



Beck-edges: -

Cross-edges: ۳-۲, ۵-۲, ۶-۴, ۳-۴

Visiting times:

۰: ۰

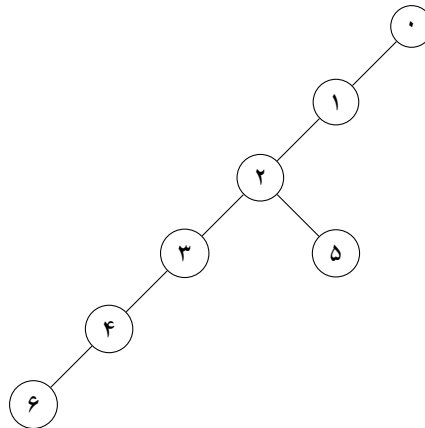
۱, ۳: ۱

۲: ۵, ۶

۳: ۴

DFS:

Tree:



Beck-edges: ۰-۳, ۱-۳, ۲-۴, ۱-۶, ۱-۵

Cross-edges: -

Starting and finishing times:

۰: ۰ to ۱۱

۱: ۱ to ۱۰

۲: ۲ to ۹

۳: ۳ to ۷

۴: ۴ to ۶

۵: ۵ to ۵

۵: ۸ to ۸

۲. تولید درخت

۱۵ نمره

گرافی با n راس و n یال داریم. الگوریتمی بهینه ارائه دهید که یالی را پیدا کند که با حذف آن، گراف تبدیل به درخت می‌شود (اگر چند یال با این ویژگی وجود داشت، می‌توانید یکی از آن‌ها را به دلخواه انتخاب کنید).

پاسخ:

از یک راس دلخواه الگوریتم dfs یا bfs را اجرا می‌کنیم و درخت حاصل از این عملیات را به دست می‌آوریم. این درخت، $n-1$ یال دارد. کافی است یالی از گراف که در درخت dfs یا bfs نیست را پیدا کنیم. این یال که یک back-edge یا cross-edge است، جواب مسئله خواهد بود. چون با حذف آن به یک درخت می‌رسیم. هم‌چنین پیچیدگی زمانی این الگوریتم برابر $O(n)$ است که نشان می‌دهد الگوریتمی بهینه ارائه داده‌ایم.

۳. سودوکوی جادویی-سرعتی

۱۵ نمره

فرض کنید یک مربع n در n به شما داده شده است. در ابتدا، در تعدادی از خانه‌های این جدول، عدد صفر نوشته شده است. شما باید تمام این خانه‌های آن را به نحوی پر کنید که:

* تمام اعداد نوشته شده، صحیح نامنفی باشند.

* تفاضل اعداد نوشته شده در دو خانه‌ی مجاور (به نحوی که یک ضلع مشترک داشته باشند) حداکثر برابر ۱ باشد.

* بزرگ‌ترین عدد نوشته شده در جدول، بیشینه شود.

برای مثال، در صورتی که تمام خانه‌های باقی‌مانده را با ۱ پر کنید، شرط اول و دوم رعایت می‌شوند. اما به احتمال زیاد شرط سوم رعایت نخواهد شد. الگوریتمی ارائه دهید که این جدول را در سریع‌ترین زمان ممکن پر کند.

پاسخ:

از نسخه‌ی multi-source الگوریتم bfs برای حل این سوال استفاده می‌کنیم. به این صورت که ابتدا در صف تمام خانه‌هایی را قرار می‌دهیم که در آن‌ها به صورت پیش فرض ۰ نوشته شده است. می‌دانیم visiting time این رئوس برابر صفر است. اگر الگوریتم bfs را برای این صف اجرا کنیم (هر بار یکی از خانه‌ها را از صف خارج کرده و همسایگانش را در صورتی که قبلاً دیده نشده‌اند، در آن درج می‌کنیم)، کافی است در هر خانه visiting time آن را بنویسیم. این الگوریتم به ما تضمین می‌دهد که ماکزیمم عدد نوشته شده در جدول (فرض کنید m)، بیشینه خواهد شد. به این معنی که نمی‌توان طور دیگری جدول را پر کرد که خانه‌ای با عدد k به طوری که از m بزرگ‌تر است داشته باشیم.

۴. ماموریت نهایی

۱۵ نمره

اخیراً منطقه‌ای عجیب در خلیج فارس کشف شده است که شامل n جزیره و m پل چوبی بین آن‌ها می‌شود که سال‌ها پیش، توسط بومی‌ها احداث شده‌اند (بین هر دو جزیره نهایتاً یک پل داریم). بعد از بازدید وزیر راه و شهرسازی از این مجمع‌الجزایر کوچک، بومیان منطقه از او خواستند که با احداث تعدادی پل جدید، امکان سفر زمینی بین تمام جزایر را برای آن‌ها فراهم سازد، به این علت که جابه‌جایی با قایق برای آن‌ها سخت است. هم‌چنین به علت کمبود منابع، دیگر خودشان امکان ساختن پل جدید را ندارند. وزیر، با توجه به وقت محدودی که دارد، با آن‌ها توافق کرد که k پل جدید احداث خواهد کرد. ممکن است تمامی جزایر به هم متصل نشوند، اما او تعداد جزایری که بین آن‌ها امکان سفر زمینی فراهم باشد را بیشینه خواهد کرد.

به وزیر کمک کنید با الگوریتمی بهینه پیش از شروع کار، محل این پل‌ها را مشخص کند.

پاسخ:

ابتدا به کمک الگوریتم‌های dfs یا bfs، مولفه‌های همبندی را بدست می‌آوریم، به این صورت که به هر راس یک label می‌دهیم (شمارش را از ۱ شروع می‌کنیم). کار از یک راس دلخواه شروع کرده و به تمام رئوس متصل به آن که طی عملیات dfs یا bfs به دست می‌آید، برچسب i می‌دهیم. سپس عملیات پیمایش را روی راس دلخواه دیگری که برچسب نخورده اجرا می‌کنیم و به رئوس به دست آمده عدد $i+1$ می‌دهیم و این عملیات را تا label خوردن تمامی رئوس ادامه می‌دهیم.

بعد گروه‌های به دست آمده را به ترتیب تعداد راس‌هایشان به صورت نزولی مرتب می‌کنیم. مجموع k عدد اول این دنباله (k جزیره‌ای که متصل کردنشان به هم منجر به رسیدن به بزرگ‌ترین شبکه‌ی زمینی ممکن می‌شود)، برابر جواب مسئله خواهد بود.

۵. رانندگی

۲۰ نمره

رایان گاسلینگ در شهر لوس آنجلس اقامت دارد. فرض کنید این شهر متشکل از n منطقه و m خیابان دو طرفه بین آن‌ها است (بین هر دو منطقه، نهایتاً یک خیابان وجود دارد). رایان که علاقه‌ی زیادی به تنهایی رانندگی کردن دارد، عادت دارد هر روز از خانه‌اش در منطقه‌ی a تا خانه‌ی دوستش در منطقه‌ی b براند. هم‌چنین فرض کنید او کوتاه‌ترین مسیر ممکن بین دو نقطه را برای رانندگی انتخاب می‌کند.

اخیراً شهردار اعلام کرده است که قصد ساختن یک خیابان جدید در شهر را دارد و از مردم خواسته که پیشنهاد‌های خود را برای محل

احداث این خیابان به او بدهند (هرکس می تواند لیستی از دوتایی های x و y به شهردار بدهد، به این معنی که علاقه مند است این خیابان جدید بین مناطق x و y احداث شود). این موضوع، رایان را به شدت نگران کرده است، به این علت که او نمی خواهد احداث این خیابان جدید، فاصله ی بین خانه اش تا خانه ی دوستش را کم تر کند (او هرچه طولانی تر رانندگی می کند، آرامش بیش تری می گیرد).

الگوریتمی از مرتبه ی زمانی $O(n^2)$ ارائه دهید به طوری که رایان گاسلینگ بتواند این لیست را آماده کرده و برای شهرداری پست کند.

پاسخ:

از دو راس a و b الگوریتم bfs را اجرا میکنیم. سپس به ازای هر دو راسی که به یک دیگر یال ندارند (مثل c و d)، فاصله ی آن ها از a و b را محاسبه و با هم جمع کرده و سپس به علاوه ی ۱ می کنیم (حاصل، فاصله ی a و b از هم است، اگر این یال وجود داشته باشد). اگر این مقدار، از فاصله ی کنونی a و b کمتر نشد، به این معنی است که یال بین c و d یک یال مناسب برای اضافه شدن است.

این کار را برای تمام رئوسی که به هم یال ندارند انجام می دهیم. در نهایت پیچیدگی زمانی آن برابر $O(n^2)$ خواهد شد.

۶. رانندگی ۲

۲۰ نمره

فرض کنید نقشه ی شهری با n میدان و $n-1$ خیابان بین آن ها به شما داده شده است به طوری که گراف حاصل، یک درخت را تشکیل می دهد. رایان گاسلینگ طی سفری که به این شهر داشت، به شدت به آن علاقه مند شد و حالا قصد دارد خانه ای در یکی از میدان های آن خریداری کرده و به اینجا نقل مکان کند. اما او مطمئن نیست با توجه به خواسته ی جدیدی که دارد، کدام میدان ها برایش بهترین گزینه محسوب می شوند. خواسته ی او به شرح زیر است:

فرض کنید او میدان x را انتخاب کرده است به طوری که اگر آن را ریشه در نظر بگیرید، ارتفاع درخت حاصل برابر h خواهد شد. رایان قصد دارد هر روز از خانه اش (در عمق صفر) تا برگگی که در عمق h دارد (مثلا راس y) رانندگی کند. هم چنین از آن جا که از رانندگی های طولانی در لوس آنجلس خسته شده، علاقه مند است که این فاصله کمینه باشد.

الگوریتمی ارائه دهید که رایان گاسلینگ در کم ترین زمان ممکن تمام میدان های مناسب را پیدا کند.

پاسخ:

برای حل این سوال، کافی است طولانی ترین مسیر موجود در درخت را به دست بیاوریم. فرض کنید نام این مسیر p و رئوس دو سر آن، a و b باشند به طوری که طول آن برابر H است. جواب مسئله، راس (یا راس های) میانی این مسیر خواهد بود.

برای اثبات این ادعا، از برهان خلف استفاده می کنیم. فرض کنید راس دیگری مانند x وجود دارد، که در صورتی که به عنوان ریشه ی درخت در نظر گرفته شود، ارتفاع حاصل برابر h خواهد شد. به طوری که $h \leq \frac{H}{4}$.

حالت ۱. فرض کنید راس x روی بلندترین مسیر گراف قرار دارد. با توجه به این که فرض کردیم این راس وسط p قرار ندارد، پس فاصله ی آن از حداقل یکی از رئوس a یا b بیشتر از $\frac{H}{4}$ است. در نتیجه ارتفاع درخت حاصل، بزرگتر از $\frac{H}{4}$ بوده (فاصله ی ریشه ی x تا مثلا برک a) که این تناقض است.

حالت ۲. فرض کنید راس x روی این مسیر قرار ندارد، و نزدیک ترین راس p به آن، y نام دارد. هم چنین فرض کنید فاصله ی این دو راس برابر k است. می دانیم فاصله ی راس y از حداقل یکی از رئوس a یا b بیشتر یا مساوی $\frac{H}{4}$ است (مثلا راس a). در نتیجه، مسیری که یک سر آن x است و از y رد می شود و به a می رسد، طولی بزرگتر از $\frac{H}{4}$ دارد. این موضوع باعث می شود درختی که ریشه ی آن x است، برگگی داشته باشد که در فاصله ای دورتر از $\frac{H}{4}$ از آن قرار دارد پس ارتفاعش بیشتر از این مقدار است و به تناقض رسیدیم.

در نتیجه غیر از نقطه (یا نقاط) میانی p راس دیگری نمی تواند مناسب رایان گاسلینگ باشد. برای پیدا کردن این مسیر هم کافی است از یک راس دلخواه dfs یا bfs بزنیم، و دو برگگی که در عمیق ترین لایه های درخت هستند را پیدا کنیم و راس میان آن ها را به عنوان جواب در نظر بگیریم.