

۹۸۴۴۴۲۲

محمد عرفان شایع زینب

سوال ۱۱

از الگوریتم BFS استفاده می‌کنیم. کوتاه‌ترین مسیر را بین دو رأس می‌یابیم که ممکن است از یال‌های بی‌وزن باشند. بنابراین کار را با الگوریتم‌های Shortest Path انجام می‌دهیم. چون وزن منفی نداریم و خود از دایره‌های هم‌گفت. اگر فرض شود حلقه‌ای با وزن کمتر وجود داشته باشد، مثلاً e هم می‌تواند $A < d(e) + w(e)$ چون یک حلقه است (دور است) می‌شود آنرا بدین گونه در نظر گرفت که $v_{i+1} \rightarrow v_n \rightarrow v_{n-1} \rightarrow \dots \rightarrow v_2 \rightarrow v_1 \rightarrow e$ در حالی که در دیدگاه اول $v_{i+1} \rightarrow v_n \rightarrow v_{n-1} \rightarrow \dots \rightarrow v_2 \rightarrow v_1 \rightarrow e \rightarrow v_{i+1}$ بود. همان طور که مشخصه عناصر دور همان یکی است و وزن حلقه (دور) تغییر نکرده و همان A است یعنی در واقع

$$A = w(v, v_{i+1}) + \dots + w(v_n, v_1) + \dots + w(v_{i+1}, v_1) + w(v_{i+1}, u) + w(u, v)$$

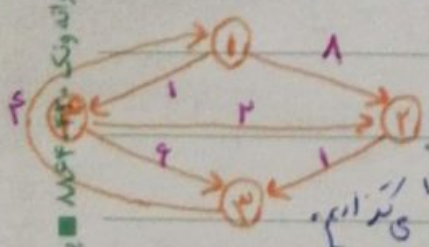
اما این بدین گونه می‌شود که

$$w(v, v_{i+1}) + \dots + w(v_n, v_1) + \dots + w(v_{i+1}, v_1) + w(v_{i+1}, u) + w(u, v) - w(e) < d(e)$$

که ما مسیر کوتاه‌تر از دایره را یافته‌ایم که با الگوریتم در تناقض است و فرض خلف بوده و حکم ثابت می‌شود.

سوال ۱۲

ابتداءً صدوق کوچک می‌کنیم که طوق باشد اگر بود سعی بر حذف یک یا چند گره بین وزن یال در گراف چنین چیزی نیست پس ادکند.



ما ترس فاصله ادس لای تویم که فاصله هر دو رأس با وزن‌های داده شده است. برای رؤس دارای یال بینشان می‌شود وزن یال و اگر یکی نباشد ∞ می‌گذاریم.

$$D = \begin{bmatrix} 0 & 1 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & \infty & 0 & \infty \\ \infty & 2 & 1 & 0 \end{bmatrix}$$

از الگوریتم Floyd warshall در بحث تغییر ماتریس استفاده می کنیم. می دانیم که این الگوریتم بدین گونه است که
 یک پیکان (k) را در نظر گرفته و بدان وسیله چک می کند که آیا با رفتن از این پیکان (k) به پیکان (i) و سپس از پیکان (i) به پیکان (j) مسیری کوتاه تر
 داریم یا خیر.

$n \leftarrow \text{rows}[w]$

for $k \leftarrow 1$ to n

do for $i \leftarrow 1$ to n

do for $j \leftarrow 1$ to n

do $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

Return d

حسب بدین وسیله ماتریس $k=|V|$ جلوی رویم و در نهایت ماتریس نهایی را می یابیم.

$$D^0 = \begin{bmatrix} 1 & 0 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 5 & 5 \\ \infty & 2 & 9 & 0 \end{bmatrix}$$

همچنین زمانی: 3×3 و در هر پیکان $O(n^2)$

همچنین حافظه: ماتریس دو بعدی $O(n^2)$

$2 \rightarrow 3: 2 \rightarrow 1 \rightarrow 3$
 $4 \rightarrow 3: 4 \rightarrow 1 \rightarrow 3$

$$D^{(1)} = \begin{bmatrix} 1 & 0 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 5 & 5 \\ \infty & 2 & 3 & 0 \end{bmatrix}$$

$2 \rightarrow 1: 2 \rightarrow 3 \rightarrow 1$

$4 \rightarrow 3: 4 \rightarrow 2 \rightarrow 3$

$4 \rightarrow 1: 4 \rightarrow 2 \rightarrow 1$

$$D^{(2)} = \begin{bmatrix} 1 & 0 & 9 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 12 & 5 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix}$$

$1 \rightarrow 3: 1 \rightarrow 2 \rightarrow 3$

$2 \rightarrow 4: 2 \rightarrow 3 \rightarrow 4$

$4 \rightarrow 1: 4 \rightarrow 2 \rightarrow 1$

$$D^{(3)} = \begin{bmatrix} 1 & 0 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 5 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix}$$

$1 \rightarrow 2: 1 \rightarrow 4 \rightarrow 2$

$2 \rightarrow 3: 2 \rightarrow 4 \rightarrow 3$

$3 \rightarrow 2: 3 \rightarrow 4 \rightarrow 2$

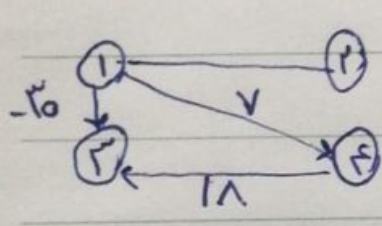
عناصر مورب ماتریس (چون طوقه نداریم) همیشه صفره چون اگر رأس داریم ماتریس مربعی مان

است و اینها ماتریس $k=4$ است.

سوال (۳)

الف) حل

چون ممکن است با افزودن مقدار سبب جابجایی کوتاه ترین مسیر شویم و چون دایمسترا فقط دردی مقدار مثبتی می شود از این نکته استفاده کرده باید مثال مناسبی داد که با مثبت کردن هزینه یال ها سبب جابجایی مسیر کوتاه ترین می شویم چنان که هزینه یال منفی داریم، دایمسترا به این توجه نمی کند و یال های مثبت یعنی یاره دل و حق



مثبت شود، در نظر می گیریم. مثال: وقتی عدد مثبت ۵ اضافه نشود اگر وقتی بخوایم از ۱ به ۳ برویم، هزینه ۵ است ولی وقتی اضافه شود هزینه ۰ بوده و مستقیم از ۱ به ۳ می رود و در حالت قبلی ۱۸ بود.

ب) (۴)

روش با الگوریتم BFS. چون اگر پیچیدگی های زمانی بررسی شود با استفاده از الگوریتم دایمسترا در صورت عدم وجود یال منفی، $O(V+E)$ است و برای الگوریتم Floyd-warshall نیز $O(V^3)$ است. همچنین با این حال که پیچیدگی BFS نیز $O(V+E)$ بیان می شود ولی بدلیل اینکه ترافیک بدون جهت باید از هر دو طرف (لبه) یال بررسی شود باید ضرب ۲ هم در واقع دارد و بدین ترتیب در اصل $O(V+2E)$ می باشد. اما در BFS چون وزن نداریم در $O(V+E)$ حل می شود بدین گونه که با BFS \hookrightarrow predecessor رأس مورد بررسی را در چک می حساب ذخیره می کنیم. حل مسیری کوتاه را می گیریم با $O(1)$ و برای بررسی وجودی حداکثر $O(V)$ لازم است. همچنین برای خود انجام الگوریتم BFS نیز $O(V+E)$ داریم که در نتیجه پیچیدگی زمانی $O(V+E)$ می شود.

پ) ۱- بله. چون ساختار مسیر در حال بهیابش عوض می شود و مسیر دیگری را انتخاب می نماید و سبب ایجاد مشکل در محاسبه shortest path می شود.

دخترانه صادقیه ۲۸۱۷ دخترانه توحید ۲۲۵۲۵۲۵ دخترانه ولی ۸۸۲۰۲۲۰ پسرانه صادقیه ۱۱۱۱۱۱۱۱

سؤال ۴۳

برای حل این شبکه، از الگوریتم را میسر (یا کمی تغییر استفاده می کنیم). بدین شکل که عمل Relax کردن
به بوسیله min در هر گره، و به کمک با صف Priority برخی بعد.

مراحل: گام اول را بر اساس یال های خروجی از هر گره (در اساس وزن) می توسیم. ترتیب ورود (برای
حالت صعودی) از کمترین وزن یال است. برای عمل کاهش ۱، بدین گونه عمل می شود که هر رأس در صف
اولویت قرار گرفته تا زمانی که در بالای صف ظاهر شده و تغییر می کند. صف اولویت با عمل یال های ورودی
در صفی گیریم.

روش Relax کردن: با Pop یال ها و چگونگی یال به پایان رسد، از صف اولویت برای تمام یال های
خروجی رأس به ترتیب افزایشی از گره گراف شروع می کنیم و وزن یال خروجی بزرگتر مساوی وزن
لبه ورودی بود یا یکی بزرگتر برای حالت افزایشی وزن (صعودی) همون بالایی باشد) یال
خروجی را به صف اولویت افزوده و موقعیت ذخیره را جابجایی می کنیم. با این روش هر یال حداقل
۱ یا ۲ در صورت آنجا که مسیر گامی گامی یا افزایشی باشد) می باشد که در آن هم (V و E) است
است چون هر رأسی را بر اساسی یال ها در صف اولویت ورود.