1. گراف G داده شده است، تعداد کوتاه‌ترین مسیرهای موجود در این گراف را بیابید.

2. الگوریتم فلوید-وارشال را به گونه‌ای تغییر دهید و روشی ارائه کنید که پس از اجرای فلوید-وارشال بتوان علاوه بر طول کوتاه‌ترین مسیر بین رئوس v و u خود مسیر را هم پیدا کرد. زمان اجرای این الگوریتم را تحلیل کنید.

3. تعریف ضرب دو ماتریس C = AB به شکل زیر است:

$$C_{ij} = \sum_{k=1}^{m} A_{ik}B_{kj}$$

- گراف بدون وزن و بدون جهت G با ماتریس مجاورت M را در نظر بگیرید، عدد واقع در $(M^K)_{ij}$ برابر با چه مقداری است؟
- آیا می‌توانید تعریف ضرب ماتریس را برای ماتریس‌های مربعی به گونه‌ای تغییر دهید که اگر W ماتریس وزن‌های یک گراف وزن‌دار جهت‌دار باشد هدد واقع در $(W^n)_{ij}$ طول کوتاه‌ترین مسیر از راس i به راس j باشد؟
- فرض کنید الگوریتمی داریم که حاصل‌ضرب بازتعریف شده در قسمت قبل را در زمان t(n) حساب می‌کند. الگوریتمی با زمان اجرای $\theta(t(n)logn)$ برای محاسبه طول کوتاه‌ترین مسیر بین تمام رئوس یک گراف ارائه دهید.

4. گراف G داده شده است، با استفاده از الگوریتم فلوید-وارشال وجود دور منفی در G را بررسی کنید.

5. گراف G داده شده است، طول کوتاه‌ترین دوری در G که تعدادی فرد یال دارد را بیابید.

۱. الگوریتم فلوید-وارشال را اینگونه تغییر میدهیم: در هر گام الگوریتم علاوه بر طول کوتاه‌ترین مسیر پیدا شده تعداد مسیرهای با این طول را نیز نگه میداریم (در ماتریس $count$). برای گام به‌روزرسانی اینگونه عمل میکنیم: اگر طول مسیری که از a به b از طریق v میرود کوتاه‌تر از مقدار فعلی بود، $count[a][b] = count[a][v] \times count[v][b]$ قرار میدهیم، اگر طول مسیر جدید مساوی مقدار قبلی بود، $count[a][b] = count[a][b] + count[a][v] \times count[v][b]$ قرار میدهیم.

۲. الگوریتم فلوید-وارشال را اینگونه تغییر میدهیم: علاوه بر ماتریس طول کوتاه‌ترین مسیر، یک ماتریس دیگر به نام $parent[u][v]$ نیز نگه میداریم. محتوای $parent[u][v]$ برابر راسی با بیشترین شماره است که کوتاه‌ترین مسیر از u به v از آن میگذرد. مقادیر اولیه این ماتریس به شکل زیر است:

$$parent[u][u] = -۱$$

$$parent[u][v] = u,\ if\ w[u][v] \neq \inf$$

برای گام به روز رسانی اگر مسیر گذرنده از راس فعلی (v) کوتاه‌تر از مسیر بدست آمده تا این مرحله بود، $parent[a][b] = v$ قرار میدهیم. برای بدست آوردن کوتاه‌ترین مسیر از راس u به v به صورت بازگشتی عمل میکنیم: ابتدا کوتاه‌ترین مسیر از u به $parent[u][v]$ را طی میکنیم، سپس کوتاه‌ترین مسیر از $parent[u][v]$ به v را. حالت پایانی زمانی‌ست که $parent[u][v] = -۱$ باشد.

۳. تعریف ضرب دو ماتریس $C = AB$ به شکل زیر است:

$$C_{ij} = \sum_{k=۱}^{m} A_{ik}B_{kj}$$

- عدد واقع در $(M^n)_{ij}$ برابر است با تعداد گشت‌های به طول n بین رووس i و j. اثبات را با استقرا انجام میدهیم. درستی پایه استقرا $n = ۱$ واضح است. برای گام استقرا داریم

$$(M^n)_{ij} = \sum_{k=۱}^{m} (M^{n-۱})_{ik}M_{kj}$$

یعنی مقدار $(M^n)_{ij}$ برابر است با حاصل‌جمع تعداد گشت‌های به طول $n - ۱$ از راس i به راس k ضربدر تعداد گشت‌های به طول ۱ از راس k به راس j به ازای تمام kهای ممکن. و این مقدار برابر است با تعداد گشت‌های به طول n بین رووس i و j.

- تعریف زیر را در نظر میگیریم:

$$C_{ij} = \min_{k=۱}^{m}\{A_{ik} + B_{kj}\}$$

ادعا میکنیم طبق این تعریف مقدار موجود در $(W^m)_{ij}$ شامل طول کوتاه‌ترین گشتی با حداکثر m یال از راس i به j است. حکم را به استقرا ثابت میکنیم. برای $m = ۱$ حکم بدیهی‌ست. برای گام استقرا داریم:

$$(W^m)_{ij} = \min_{k=۱}^{n}\{(W^{m-۱})_{ik} + W_{kj}\}$$

یعنی مقدار $(W^m)_{ij}$ برابر است با حداقل طول گشتی که از i با گشتی با حداکثر m یال به یک راس دیگر مثل k میرود و سپس با یک یال از k به j میرود (اگر $k = j$ مقدار همان طول کوتاه‌ترین گشت به طول حداکثر $m - ۱$ یال میشود). مقداری که ذکر برابر با مقدار ادعا شده‌است.

- کافیست مقدار $W^{\lceil logn \rceil}$ را حساب کنیم که با $\lceil logn \rceil$ بار به توان ۲ رساندن مکرر محاسبه میشود.

۴. ادعا میکنیم گراف دور منفی دارد اگر و تنها اگر پس از اجرا الگوریتم فلوید-وارشال اگر مقدار $distance[v][v]$ به ازای حداقل یک راس منفی باشد. یک سمت ادعا واضح است، در صورتی که چنین راسی موجود باشد گراف دور منفی دارد. از سمت دیگر، یک دور منفی در گراف را در نظر میگیریم. دنباله رووس موجود در این دور را $v_۱ \dots v_k$ مینامیم به گونه‌ای که v_k راس با بیشترین شماره است. هنگامی که در زمان اجرای الگوریتم فلوید-وارشال راس میانی برابر v_k است، مقدار $distance[v_۱][v_{k-۱}]$ حداکثر برابر طول مسیر $v_۱ \dots v_{k-۱}$ است، سپس با به‌روزرسانی از طریق v_k مقدار $distance[v][v]$ با توجه به منفی بودن طول دور یک عدد منفی میشود.

۵. ابتدا طول کوتاه‌ترین مسیرهای با زوج یال و فرد یال را بین هر جفت راس پیدا میکنیم. برای این کار الگوریتم فلوید-وارشال را اینگونه تغییر میدهیم: بجای نگه داشتن یک ماتریس شامل طول کوتاه‌ترین مسیرهای بین رووس تا مرحله فعلی، یک ماتریس شامل طول کوتاه‌ترین مسیر با زوج یال و یک ماتریس شامل طول کوتاه‌ترین مسیر با فرد یال نگه میداریم، برای اشاره به این دو ماتریس به ترتیب $sep[u][v]$ و $sop[u][v]$ را استفاده میکنیم. مقادیر اولیه این دو ماتریس به شکل‌های زیرند:

$$sop[u][v] = w[u][v] \quad , if\ u \neq v, sop[u][u] = \inf$$

$$sep[u][v] = \inf \quad , if\ u \neq v, sep[u][u] = ۰$$

برای گام بروز رسانی اینگونه عمل میکنیم:

$$sop[a][b] = min(sop[a][b], sep[a][v] + sop[a][v], sop[a][v] + sep[a][v])$$

$$sep[a][b] = min(sep[a][b], sep[a][v] + sep[v][b], sop[a][v] + sop[v][b])$$

سپس طول کوتاه‌ترین دور فرد به شکل زیر پیدا میشود:

$$\min_{u,v,w \in V(G)} (sop[u][v] + sop[v][w] + sop[w][u], sep[u][v] + sep[v][w] + sop[w][u])$$

6. موارد موجود در گروه الف را به مورد مرتبط با آن‌ها در گروه ب وصل کنید.

گروه الف)

1) الگوریتم دایکسترا
2) الگوریتم بلمن فورد
3) الگوریتم فلوید وارشال

گراف ب)

1) Dynamic Programming
2) backtracking
3) greedy algorithm

جواب:

دایکسترا یک الگوریتم حریصانه (greedy algorithm) است.

بلمن فورد و فلوید وارشال هر دو Dynamic Programming هستند.

7.یک گراف جهتدار G=(V,E)بما داده شده که در ان از هر یال E∍(U,V) همراه یک مقدار R(U,V) که یک عدد حقیقی در محدوده [1,0]است میباشد.این عدد نشان دهنده قابلیت اطمینان از یک کانال ارتباطی از راس Uبه V است با فرض استقلال این احتمالات و اینکه R(V,U)احتمال شکست را مشخص کند یک الگوریتم کارامد برای یافتن قابل اطمینان ترین مسیر بین هر دو راس داده شده چیست؟

برای حل این سوال نیاز داریم مسیری با مینیمم ضرب یالها را بیابیم فرض کنیم s مبدا و tراس مقصد باشد

$$P = \arg max \prod_{i=0}^{k} r(v_{i-1}, v_i)$$

این سوال به راحتی به یک مسأله کوتاهترین مسیر از یک رأس تبدیل میشود .پیش از ان لازمست وزن هر راس را به $-\log r(u,v)$ تبدیل کنیم از انجا که لگاریتم یکنواختی را تغییر نمیدهد و منفی لگاریتم در این وزنهای جدید باعث میشود مسأله بجای کوچک کردن به بزرگ کردن تبدیل شود در نتیجه

$$P = \arg\min \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

کم وزنترین مسیر را در گراف اولیه خواهد داد که با استفاده از Dijkstraقابل حل خواهد بود و در $O(E\log V)$قابل انجام است

8. راه دیگر برای بازسازی کوتاه ترین مسیر در الگوریتم فلوید-وارشال با استفاده از مقادیر $\Phi_{ij}^{(k)}$است برای هر i,j,k=1,2,3....n مقدار $\Phi_{ij}^{(k)}$ بیشترین عدد داده شده بین راسهای میانی کوتاهترین مسیر بین راس i و jاست و همه راسها از مجموعه {1,2,....k}میباشد یک فرمول بازگشتی از فلوید وارشال ارایه دهید که مقادیر $\Phi_{ij}^{(k)}$ را محاسبه و چاپ کند و ماتریس($\varphi = (\Phi_{ij}^{(k)}$ را بعنوان ورودی بگیرد.

ابتدا باید بدانیم که $\Phi_{ij}^{(k)}$ پدر راس jدر کوتاهترین مسیر از iتا j را ذخیره میکند

$$\Phi_{ij}^{(k)} = \begin{cases} NIL & if\ k = 0 \\ k & if\ k > 0\ \ and\ \ d_{ik}^{k-1} + d_{kj}^{k-1} < d_{ij}^{k-1} \\ \Phi_{ij}^{(k)} & if\ k > 0\ \ and\ \ d_{ik}^{k-1} + d_{kj}^{k-1} \geq d_{ij}^{k-1} \end{cases}$$

Algorithm 1: FLOYD-WARSHALL(W)

$n = W.rows$;
$D^{(0)} = W$;
$\Phi^{(0)} = NIL$;
for $k = 1$ *to* n **do**
  let $D^{(k)} = (d_{ij}^{(k)})$ be a new $n \times n$ matrix;
  let $\Phi^{(k)} = (\phi_{ij}^{(k)})$ be a new $n \times n$ matrix;
  for $i = 1$ *to* n **do**
    for $j = 1$ *to* n **do**
      if $d_{ik}^{(k-1)} + d_{kj}^{(k-1)} < d_{ij}^{(k-1)}$ **then**
        $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$;
        $\phi_{ij}^{(k)} = k$;
      end
      else
        $d_{ij}^{(k)} = d_{ij}^{(k-1)}$;
        $\phi_{ij}^{(k)} = \phi_{ij}^{(k-1)}$;
      end
    end
  end
end

Algorithm 2: PRINT-ALL-PAIRS-SHORTEST-PATH(Φ, i, j)

```
if i == j then
    print i;
end
else if φ_ij == NIL then
    print "no path from " i " to " j " exists".;
end
else
    PRINT-ALL-PAIRS-SHORTEST-PATH(Φ, i, φ_ij);
end
```

9.فرض کنیم $G=(V,E)$ یک گراف وزندار با تابع وزنهای مثبت باشد به تعریف $w=E\rightarrow\{1,2,\dots,W\}$ برای مقادیر مثبت W است و فرض کنیم هیچ دو راسی کوتاهترین مسیر با وزن یکسانی را تا راس مبدا sندارند .حال فکرکنید گرافی بدون وزن به نام $G'=(V\cup V',E')$ را تعریف کنیم که از جا گذاری هر یال $(u,v)\in E$ با $w(u,v)$ وزن واحد در مجموعه ریوس بدست میاید G' چند راس دارد ؟حال فرض کنید که BFS روی G' اعمال میکنیم ثابت کنید اردر این BFS درسیاه کردن ریوس برابر با اردر دایکسترا در خارج کردن انها از صف اولویت اعمال شده بر G است.

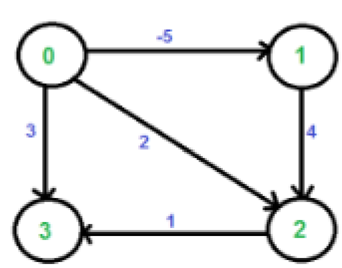
میدانیم G' دارای مجموعه ای از ریوس $V\cup V'$ است برای هر راس (u,v) در G به دلیل انکه یالهای $w(u,v)$ یال با وزن واحد در G' وجود دارد در نتیجه $w(u,v)-1$ راس داخلی در G' وجود خواهد داشت .در نتیجه مجموع ریوس داخلی در G'

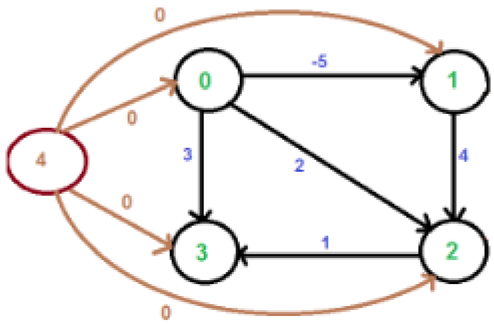
$$|V'| = \sum_{(u,v)} (w(u,v) - 1) = \sum_{(u,v)} w(u.v) - |E|$$

$$|v \cup v'| = |V| + |V'| = \sum_{(u,v)\in E} w(u,v) + |V| - |E|$$

حال فرض کنید v از صف اولویت اکسترکت شده صف=Q.با علم به اینکه هیچ دو راسی وزنهای کوتاهترین مسیرشان یکسان نیست درنتیجه $d[u]<d[v]$.در G' بدلیل اینکه هر راس (u',v') در G' به اندازه $w(u',v')$ بزرگ شده اند در BFS ان در مرحله discovery ان u ویزیت شده ودر مرحله V $d[v]$ ویزیت شده است در نتیجه v بعد از u ویزیت شده .بعبارت دیگر در BFS v بعد از U ویزیت شده پس عمق v از عمق u بیشتر است که نشان میدهد $d[v]>d[u]$ درنتیجه در دایکسترا نیز پس از u خارج میشود.

10.الگوریتم جانسون را بر گراف زیر اعمال کنید :

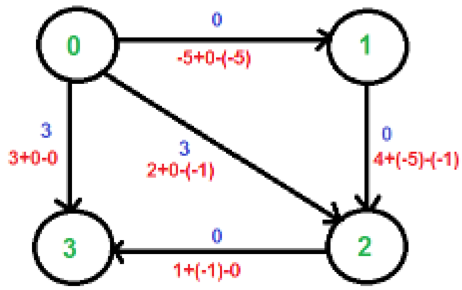




1. راس 4را اضافه میکنیم.
2. با بلمن فورد کمترین فاصله راس 4 را از همه ریوس حساب میکنیم و همه ریوس را به $w(u, v) = w(u, v) + h[u] - h[v]$. اپدیت میکنیم سپس روی هر راس دایکسترا اعمال میکنیم

هزینه نهایی:

$O(V^2\log V + VE)$. است .



11.اگر G_s یک گراف احتمالات باشد محتملترین مسیر بین هر دو راس را بایک الگوریتم تعیین کنید:

1.راسهای با وزن صفر به G_sاضافه میکنیم تا گراف کامل شود

2.ماتریس $W=[w_{ij}]$را ماتریس وزن یالها تعیین میکنیم و وزن یالها دران ذخیره میشود.

3.ماتریس پدر $T=[t_{ij}]$رابا تعداد یکسان ردیف و ستون میسازیم و به این شکل مقدار دهی میکنیم :

$$t_{ij} = \begin{cases} 0, i = j \\ i, i \neq j \end{cases} =$$

هر راس رو اپدیت میکنیم (*relax*)بشکل زیر:

اگر $w_{ij} < w_{ik}.w_{kj}$ انگاه $t_{ij} = t_{jk}$

$$w_{ij} = w_{ik}.w_{kj}$$

ماتریس نهایی W ماتریس احتمال از محتملترین دنباله اتفاقات است (w_{ij}احتمال محتمل ترین دنباله است که ازاتفاق iشروع و به jختم میشود)