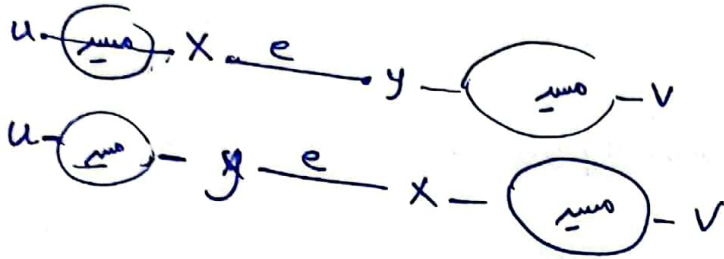


① برای هر مسیری بین دو راس u, v می‌گوئیم که فزین می‌سین دوسریال e راس‌های x و y

هستند (x, y) حال اگر مسیر بین u, v از یالین e بگذرد خواهیم داشت که:

دو حالت:



۱

۲

حال باین فرض برای هر دوراس u, v می‌خواهیم اولی

$$\text{if } (disc[u][v] > disc[u][x] + disc[y][v] + w_e)$$

$$disc[u][v] = disc[u][x] + disc[y][v] + w_e$$

$$\text{else if } (disc[u][v] > disc[u][y] + disc[x][v] + w_e)$$

$$disc[u][v] = disc[u][y] + disc[x][v] + w_e$$

② اگر هر از راس برابر یک بود در نظر بگیریم w تنج‌های تبدیل راس‌ها بگیریم مثلاً اگر نرخ تبدیل

از A به B از B برابر x است یال از A به B با وزن x داریم در نظر بگیریم

حال ابتدا باید به این نکته توجه کرد که اگر ما n تبدیل از روی داشته باشیم (که البته به اعتبار داریم هم جز

آن نیست) در صورتی سود کرده ایم که این تبدیل‌های از روی به ما سبب از مقدار اولیه پول دهند یعنی رابطه

$$w_1 \times w_2 \times w_3 \times \dots \times w_n > 1$$

زیر برقرار باشد.

$$\log w_1 + \log w_2 + \dots + \log w_n > 0 \quad (\log 1 = 0)$$

اگر از این رابطه \log بگیریم داریم:

$$-\log w_1 - \log w_2 - \dots - \log w_n < 0$$

اگر در منفی ضرب شوند داریم که

از طرفی می‌دانیم که تبدیل‌های هم‌کثرت ما یک دور (cycle) بوجود می‌آورد زیرا مبدأ و مقصد یکی است

از اطلاعات بالا استفاده کرده و الگوریتم زیر را ارائه می‌کنیم.

۱) ابتدا تمام یال‌های گراف را ایدیت می‌کنیم. به این صورت که هر یال با وزن w مقدار $\log(w)$ را می‌گیرد. $O(|E|)$

۲) حال بر اساس ~~فرمول~~ نامساوی صفحه قبل با سبب برای اینکه مسئله جواب داشته باشد یک دور منفی (negative cycle) داشته باشیم در این صورت یک پاسخ داریم و اگر دور منفی نداشته باشیم پاسخ ندارد.

برای پیدا کردن دور منفی می‌توانیم از الگوریتم Bellman-Ford استفاده کنیم.

به این صورت که اگر $V-1$ بار راس داریم ۱-۱۷۱ بار یال‌ها را Relax می‌کنیم.

و در آخر یکبار Relax می‌کنیم و اگر در این بار راس مقدارش عوض شد (Relax) شد

$$O(|E|V^2)$$

یک دور منفی داریم!

④ مسیرهای آسفالت را در گراف مربوط (G) به ریشه می‌کنیم. دوکپی از گراف ایجاد می‌کنیم نام آنها را G' و G'' می‌گذاریم. سه گراف G و G' و G'' را به نحوی می‌کنیم یال‌های سبز گراف G' را حذف می‌کنیم یال‌های سبز در گراف‌های G و G'' را به نحوی می‌کنیم که به راس متناظر خود (راس مقصد) در گراف G و G'' (به ترتیب) وصل شوند. (حاله G) به این نحو تمام مسیرهایی که در یال‌ها سبز دارند به G' می‌روند و اگر یک یال سبز دارند به G'' می‌روند. در حال حاضر که کوتاه‌ترین مسیر از راس مبدأ را به سایر راس‌ها که شامل همه راس‌های گراف‌های G و G'' را پیدا کنیم. که می‌توان با الگوریتم دانلیستر آن را پیدا کنیم.

پیدا کردن کوتاه‌ترین مسیر از راس مبدأ $O(|E| + V \log V)$ زیرا تعداد راس‌ها ۳ برابر شده

و تعداد یال‌ها نیز می‌تواند تقریباً سه برابر $|E|$ باشد ($|E'| \leq 3|E|$)

(3)

کامیت که الگوریتم دانستیرا را اجرا کنیم و برای هر مسیر ماکسیم یال را نگه داریم و
برای هر راس ماکسیم یال مسیرش را ذخیره کنیم. در انتها (وقتی توان) ماکسیم
یال هر مسیر را برگشت آوریم.

Dijkstra-New (G, S)

Create vertex set Q , create a dictionary max-edge for each node

for each vertex v in G :

$dist[v] = \infty$

$prev[v] = \text{UNDEFIND}$

add v to Q :

$max_edge[v] = \text{UNDEFIND}$

$dist[S] = 0$

$max_edge = 0$

while Q is not EMPTY:

$u = \text{vertex in } Q \text{ with min distance}$

remove u from Q

for each adj v of u :

$a = dist[u] + w(u, v)$

if $a < dist[v]$

$dist[v] = a$

$prev = u$

$max_edge[u] = \max(w(u, v), max_edge[u])$

return $dist, Prev, max_edge$

و کامیت که ماکسیم را در max-edge برگردانیم.

Dijkstra (G, s) \rightarrow create vertex set Q :

for each vertex v in G :

$\text{dist}[v] = \text{INFINITY}$

$\text{prev}[v] = \text{UNDEFIND}$

add v to Q

$\text{dist}[s] = 0$

while Q is not EMPTY:

$u = \text{vertex in } Q \text{ with min distance}$
remove u from Q

for each adj v of u

$a = \text{dist}[u] + w(u, v)$

if $a < \text{dist}[v]$

$\text{dist}[v] = a$

$\text{prev} = u$

return dist, Prev

⑤

هر تقاطع را به عنوان یک Node در نظر می گیریم و زمان عبور از هر ضلع را برابر وزن یال

در نظر می گیریم. حال کوتاه ترین مسیر از تقاطع A (پایه آن ها) به بقیه تقاطع ها را محاسبه می کنیم.

برای این کار از الگوریتم دانستیرا استفاده می کنیم. در هر راس به این صورت بقیه راس را Relax

می کنیم که زمانی که تا آن راس سیر کردیم رادر نظر می گیریم و برای هر راس مجاور می گیریم اگر کوتاه تر از آن یال که داریم تقاطع بعدی چراغ قهوه است یا سبز و بر اساس آن به میسیم زمان در تقاطع که

چراغ سبز است می رود. در واقع حال هزینه را وزن یال + زمان بقیه تا چراغ سبز شود در نظر می گیریم

و اگر چراغ سبز باشد این زمان 0 است. همچنین هزینه راس A را برابر با مدت زمانی که ماند تا چراغ سبز شود گرفته

و بقیه را مقدار اولی می ضایت داده و مانند دانستیرا سیر می کنیم تا همسایه های هر راس را به ترتیب Relax کنیم (همه ی

که نیستیم هر راس که از minheap حذف می کنیم اگر چراغ سبز بود همسایه ها را Relax می کنیم و اگر قرمز بود.

یک پشته از زمان تا سبز شدن باقی دارد که هدف زمان تا سبز شدن را با هزینه فعلی جمع کرده و به عنوان هزینه جدید آن در نظر می گیریم و آن را در minheap اضافه می کنیم. تا زمانی که minheap خالی شود.