

## گزارش پروژه DP

### اعضای گروه :

طاہرہ فہیمی 9539045

سینا فراہانی 9525038

مہدی صابر 9525027

(A) نحوه شکستن مسالہ بدین صورت است کہ در ہر مرحلہ تمام حالت ہایی کہ بہ شہر ہای دیگر سفر میکنیم را در نظر میگیریم حال بہ صورت بازگشتی ہزینہ سفر بہ ان شہر را با ہزینہ شروع مجدد مسافرت از شہری کہ بہ ان رسیدہ ایم را در نظر میگیریم بہ عنوان مثال فرض میکنیم کہ در شہر الف قرار داریم حال از این شہر می توان بہ شہرہای ب و ث رفت حال بہ صورت بازگشتی تابع رفتن بہ یک شہر دیگر را روی شہر ب صدا میزنیم و خروجی این تابع را با ہزینہ سفر از الف بہ ب جمع می کنیم بدین صورت می توان مسالہ را مدل سازی کرد.

حال برای این کہ بتوان کم ترین مقدار مورد نظر برای این مسالہ را پیدا کرد کافیست کہ قبل از جمع کردن این دو مقدار از کمینہ بودن مقدار تابع بازگشتی مان اطمینان حاصل کنیم ( یعنی باید تابع بازگشتی مان کمترین مقدار را برای ما برگرداند و از ان جایی کہ این تابع در ہر مرحلہ مینیمم را می دہد پس بدیہی است کہ برای بالاترین سطح ہم کمترین ہزینہ را دادہ است)

برای سادہ تر شدن مسالہ اول فرض میکنیم کہ نقطہ شروع را می دانیم بہ عبارت دیگر فرض میکنیم یک شہر خاصی برای شروع ما وجود دارد. حال شبہ کد پیدا کردن راہ حل بہینہ بہ صورت زیر است:

For each CITY in cities :

$$\text{CostOfTravel} \leftarrow \text{Cost of operation (CITY)} + \text{minimum} \{ \text{cost of moving( new city)} + \text{CostOfTravel(CITY)} \}$$

برای اطمینان از این کہ الگوریتم مورد نظر مان ہمیشہ بہینہ ترین جواب را برمی گرداند ہمیشہ کم ترین مسیر مشخص شدہ تا ان شہر را ذخیرہ می کنیم تا در نہایت حالت بہینہ را بدست آوریم. در کد اصلی ما cost همان ماتریسی است کہ کمترین ہزینہ را نگہداری میکند.

(B) در مسالہ دادہ شدہ k تعداد شہر ہا و n تعداد ماہ ہایی کہ داریم است ابتدا شبہ کد مسالہ را می نویسیم سپس از روی شبہ کد بہ محاسبہ پیچیدگی زمانی این الگوریتم می پردازیم:

For each CITY in cities :

$$\text{CostTravel(CITY)} \leftarrow \text{Cost of operation (CITY)}$$

For MONTH  $\leftarrow 1$  to number of months :

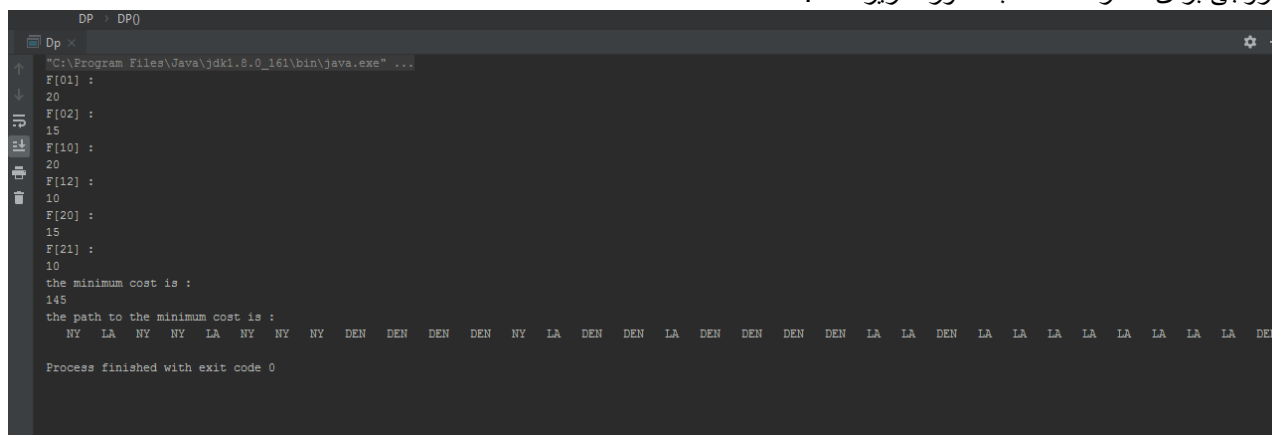
For NEXTCITY  $\leftarrow 1$  to all number of cities:

$$\text{CostTravel(CITY,MONTH)} \leftarrow \text{Cost of operation(CITY,MONTH)} + \text{minimum} \{ \text{cost of moving(NEXTCITY,CITY)} + \text{CostTravel(NEXTCITY,MONTH-1)} \}$$

For اولیه روی تمام شهرها می توانند گردش کند. از آن جایی که کمترین هزینه یک سفر مربوط به هزینه ماندن در یک شهر است پس در خط دوم باید هزینه هر سفر به ازای هر شهر را برابر هزینه ماندن در آن شهر قرار دهیم. For دوم روی تمام ماه های سال گردش میکند و For بعدی نشان دهنده شهری است که در هر مرحله باید به آن برود. چون کد بالا از سه تا for تو در تو ساخته شده است پس پیچیدگی زمانی برای این الگوریتم برابر است با  $O(k*k*n)$  خواهد بود

(C) کد برنامه پیوست شده است

(D) خروجی برای مقدار داده شده به صورت زیر است:



```
Dp < DP0
"C:\Program Files\Java\jdk1.8.0_161\bin\java.exe" ...
F[01] :
20
F[02] :
15
F[10] :
20
F[12] :
10
F[20] :
15
F[21] :
10
the minimum cost is :
145
the path to the minimum cost is :
NY LA NY NY LA NY NY NY DEN DEN DEN DEN NY LA DEN DEN LA DEN DEN DEN DEN LA LA DEN LA LA LA LA LA LA LA LA DEN
Process finished with exit code 0
```