تمرین سوم درس تحلیل و طراحی الگوریتم

مریم سادات هاشمی
سهیل رستگار
سید صالح اعتمادی

دانشگاه علم و صنعت ـ نیمسال دوم ۹۸ـ۹۷

لطفا به نکات زیر توجه کنید:

- مهلت ارسال این تمرین شنبه ۱۸ اسفند ماه ساعت ۱۱:۵۹ ب.ظ است.
- این تمرین شامل سوال های برنامه نویسی می باشد، بنابراین توجه کنید که حتماً موارد خواسته شده را رعایت کنید.
- نام شاخه، پوشه و پول ریکوست همگی دقیقا "$A3$" باشد.
- در صورتی که به اطلاعات بیشتری نیاز دارید می توانید با ایدی تلگرام @maryam_sadat_hashemi یا @soheil_rategar در ارتباط باشید.
- اگر در حل تمرین شماره ی ۳ مشکلی داشتید،لطفا به این لینک مراجعه کنید و زمانی را برای رفع اشکال تنظیم کنید.

موفق باشید.

توضیحات کلی تمرین

تمرین این هفته ی شما،۴ سوال دارد که باید به همه ی این سوال ها پاسخ دهید. برای حل این سری از تمرین ها مراحل زیر را انجام دهید:

۱. ابتدا مانند تمرین های قبل، یک پروژه به نام $A3$ بسازید.

۲. کلاس هر سوال را به پروژه ی خود اضافه کنید و در قسمت مربوطه کد خود را بنویسید. هر کلاس شامل دو متد اصلی است:

متد اول: تابع solve است که شما باید الگوریتم خود را برای حل سوال در این متد پیاده سازی کنید.

متد دوم: تابع process است که مانند تمرین های قبلی در TestCommon پیاده سازی شده است. بنابراین با خیال راحت سوال را حل کنید و نگران تابع process نباشید! زیرا تمامی پیاده سازی ها برای شما انجام شده است و نیازی نیست که شما کدی برای آن بزنید.

۳. اگر برای حل سوالی نیاز به تابع های کمکی دارید؛ می توانید در کلاس مربوط به همان سوال تابع تان را اضافه کنید.

اکنون که پیاده سازی شما به پایان رسیده است، نوبت به تست برنامه می رسد. مراحل زیر را انجام دهید.

۱. یک UnitTest برای پروژه ی خود بسازید.

۲. فولدر TestData که در ضمیمه همین فایل قرار دارد را به پروژه ی تست خود اضافه کنید.

۳. فایل GradedTests.cs را به پروژه ی تستی که ساخته اید اضافه کنید. توجه کنید که مانند تمرین های قبل، لازم نیست که برای هر سوال TestMethod بنویسید. تمامی آنچه که برای تست هر سوالتان نیاز دارید از قبل در این فایل برای شما پیاده سازی شده است.

دقت کنید که TestCommon تغییر یافته است. بنابراین شما باید نسخه ی جدید آن را با دستورGit Pull دریافت کنید .

در این تمرین، علاوه برفایل هایی با پسوند txt که تست کیس های سوالات هستند و شما از آن ها برای تست کدتان استفاده می کنید، فایل هایی با پسوند webgraphviz نیز در پوشه TestData وجود دارد. شما با استفاده از این فایل ها می توانید گراف های کوچکتر از ۱۰۰ گره را به صورت تصویری در این سایت مشاهده کنید.

۱ Computing the Minimum Cost of a Flight

دراین مسئله، ماموریت شما حداقل کردن هزینه ی پرواز است. برای این کار باید یک گراف جهت دار بسازید که در آن وزن یال بین دو گره(شهر) متناظر با هزینه پرواز بین آن دو شهر است.

اکنون در یک گراف جهت دار با یال هایی با وزن مثبت و با n راس و m یال، وزن کوتاه ترین مسیر بین u و v را پیدا کنید. (در واقع حداقل وزن کل یک مسیر از u به v)

خط اول فایل ورودی، تعداد راس های گراف(یعنی n) را مشخص می کند. هر یک از خطوط بعدی، شامل دو راس است که بدین معنی است که از راس اول به راس دوم یال وجود دارد. در خط آخر هم راس های u و v قرار دارد که الگوریتم شما باید حداقل وزن یک مسیر از u به v را پیدا کند یا اگر مسیری وجود نداشت ۱ـ چاپ کند.

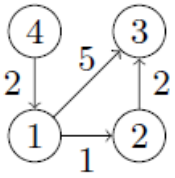
نمونه ۱

ورودی:

```
4
1 2 1
4 1 2
2 3 2
1 3 5
1 3
```

خروجی:

```
3
```



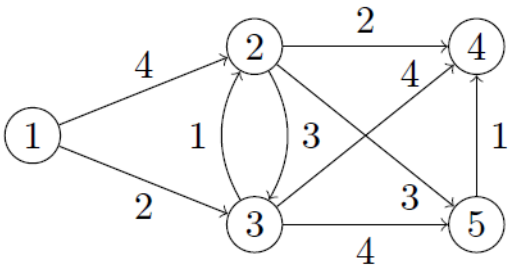
در این گراف کوتاه ترین مسیر از راس ۱ به راس ۳ (۱ -> ۲ -> ۳) با وزن ۳ است.

نمونه ۲
ورودی:

```
5
1 2 4
1 3 2
2 3 2
3 2 1
2 4 2
3 5 4
5 4 1
2 5 3
3 4 4
1 5
```

خروجی:

```
6
```

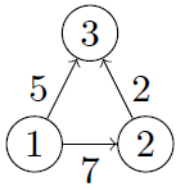


نمونه ۳
ورودی:

```
3
1 2 7
1 3 5
2 3 2
3 2
```

خروجی:

```
-1
```



۲ Detect- Anomalies in Currency Exchange Rate ing

در این سوال به شما لیستی از ارز ها $c_1، c_2، ...، c_n$ با لیستی از نرخ های مبادله داده می شود به طوری که r_{ij} تعداد واحد ارز c_j است که به ازای هر واحد c_i می توان دریافت کرد. شما باید بررسی کنید که آیا امکان دارد با شروع از یک واحد از یک ارز مشخص، دنباله ای از تبدیل هارا انجام داد و به بیش از یک واحد از ارز اولیه رسید یا خیر. برای این منظور شما باید گرافی را بسازید که راس ها، همان ارز های موجود $c_1، c_2، ...، c_n$ باشند و وزن یک یال از c_i به c_j برابر با $-log_{r_{ij}}$ باشد. و سپس تنها کافی است بررسی کنید که آیا یک دور منفی در این گراف وجود دارد یا خیر.

خط اول فایل ورودی، تعداد راس های گراف(یعنی n) را مشخص می کند. هر یک از خطوط بعدی، شامل دو راس و یک وزن است که نشان می دهد که یالی از راس اول به راس دوم با وزن مشخص شده وجود دارد. اگر در گراف دوری با وزن منفی وجود داشت در خروجی ۱ و در غیر این صورت ۰ چاپ شود.

نمونه ۱

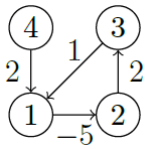
ورودی:

```
4
1 2 -5
4 1 2
2 3 2
3 1 1
```

خروجی:

```
1
```

وزن دور ۱ -> ۲ -> ۳ برابر با ۲ـ است که منفی می باشد.



۳ Exchanging Money Optimally

حال شما باید یک راه بهینه را برای تبدیل ارز c_i به تمام ارز های دیگر محاسبه کنید. با دریافت کردن یک گراف جهت دار با وزن هایی که ممکن است منفی باشند و n راس و m یال و همینطور راس s، طول کوتاه ترین مسیرها را از s به تمامی راس های گراف محاسبه کنید.

خط اول ورودی، تعداد راس های گراف را مشخص می کند. هر یک از خطوط بعدی، شامل دو راس و یک وزن است که نشان می دهد که یالی از راس اول به راس دوم با وزن مشخص شده وجود دارد. در پایان شماره ی راس s می آید. در خروجی به ازای هریک از راس های گراف به شیوه ی زیر عمل کنید: "*" چاپ شود، اگر مسیری از s به u وجود ندارد. " _ " چاپ شود، اگر مسیری از s به u وجود دارد، اما کوتاه ترین مسیر وجود ندارد (فاصله این دو راس ∞-است). در غیر اینصورت، طول کوتاهترین مسیر چاپ شود

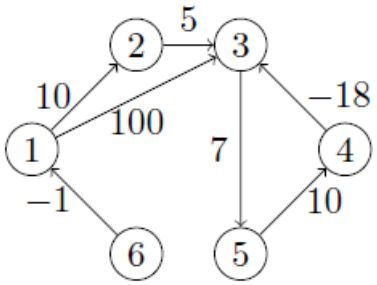
نمونه ۱

ورودی:

```
6 7
1 2 10
2 3 5
1 3 100
3 5 7
5 4 10
4 3 -18
6 1 -1
1
```

خروجی:

```
0
10
-
-
-
```



اولین خط خروجی نشان می دهد که فاصله ی ۱ تا ۱ برابر ۰ است. خط دوم نشان می دهد که فاصله ی ۱ تا ۲ برابر ۱۰ می باشد (مسیر متناظر ۱ -> ۲ است). سه خط بعدی نشان می دهند که فاصله ۱ تا راس های ۳،۴ و ۵ برابر با ∞- است. در واقع ابتدا می توان به راس ۳ توسط یال های ۱ -> ۲ -> ۳ رسید و سپس طول یک مسیر را به دلخواه با پیمودن متعدد دور ۳ -> ۵ -> ۴ که وزن منفی دارد، کوچک کرد. خط آخر نیز نشان می دهد که مسیری از ۱ به ۶ در این گراف وجود ندارد.

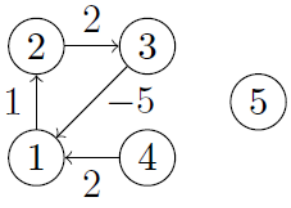
نمونه ۲

ورودی:

```
5 4
1 2 1
4 1 2
2 3 2
3 1 -5
4
```

خروجی:

```
-
-
-
0
```



۴ Friend Suggestion

شبکه های اجتماعی در ارتباط بین افراد نقش موثری دارد. بنابراین پیشنهادات دوست یکی از مهم ترین ویژگی های شبکه ی اجتماعی مثل فیس بوک است. یکی از مهمترین ورودی های الگوریتم پیشنهاد دوست، فاصله فعلی بین شما و شخص پیشنهاد شده در نمودار ارتباطات دوستان است. وظیفه شما این است که الگوریتم بهینه ای را برای بدست آوردن این فاصله طراحی کنید.

خط اول فایل ورودی شامل دو عدد صحیح n و m به ترتیب تعداد گره ها و لبه ها در شبکه است. گره ها از ۱ تا n هستند. هر کدام از خطوط بعدی شامل سه عدد صحیح u ، v و l که نشان دهنده ی یک لبه به طول l از شماره گره u به شماره گره v است. (توجه داشته باشید که بعضی از شبکه های اجتماعی توسط گراف های جهتدار نشان داده می شوند، در حالی که برخی دیگر توسط گراف های غیر جهتدار نشان داده می شوند. به عنوان مثال، توییتر یک گراف جهتدار است (به این معنی است که u دنبال کننده v است)، در حالی که فیس بوک گراف غیر جهتدار است (به این معنی است که u و v دوست هستند).

در فایل ورودی بعد از لبه های گراف، خط بعدی شامل عدد صحیح q است که تعداد کوئری ها را مشخص می کند. هر یک از خطوط بعدی نشان دهنده ی کوئری ها هستند که شامل دو گره u و v هست که شما باید فاصله ی این دو را بدست آورید و اگر هیچ مسیری بین این دو گره وجود نداشت عدد ـ۱ را برگردانید.

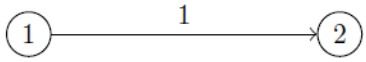
نمونه ۱

ورودی:

```
2 1
1 2 1
4
1 1
2 2
1 2
2 1
```

خروجی:

```
0
0
1
-1
```



نمونه ۲

ورودی:

```
4 4
1 2 1
4 1 2
2 3 2
1 3 5
1
1 3
```

خروجی:

```
3
```