

پروژه درس ریاضیات گسسته

استاد درس:

دکتر انتظاری

حل تمرین:

امیرحسین احمدی، سهراب نمازی نیا، سجاد رضانی، مسعود گلستانه،

رضا موسوی نیا، هادی شیخی

مقدمه

هدف ما از انجام این پروژه پیمایش و ساخت گراف است. برای اینکه اندکی با ساختار داده ای گراف آشنا شوید یک مثال خلاصه از کلاس های گراف آورده شده است.

• کلاس Node

این کلاس نشان دهنده هر راس از گراف است که ویژگی ها زیر را دارد:

- Data که همان اطلاعات مربوط به هر راس در آن ذخیره میشود.
- List<Node> که همسایه های هر راس در این قسمت نگه داری میشود.

• کلاس Graph

این کلاس کلیه Node ها را برای انجام محاسبات نگه میدارد.

حال با توجه به اطلاعات بالا بخش های پروژه را حل کنید.

برای ذخیره کردن اطلاعات گراف اختیار با خودتان است که چگونه دیتا را ذخیره کنید. نکات بالا صرفا راهنمایی است.

چند نکته:

❖ هرگونه مشابهت در کد دانشجویها به عنوان تقلب محسوب شده و نمره هر دو دانشجو صفر لحاظ خواهد شد.

❖ به هر چهار زبان C#, C++, python, java میتوانید پیاده سازی خود را انجام دهید.

❖ در این پروژه بحث تایم و performance نیست و صرفا کد شما باید صحیح باشد.

❖ سعی بر این بوده که دیتای ورودی این پروژه واقعی باشد به همین خاطر دیتا فرم خاصی دارد. برای مثال داده ورودی اینگونه است:

IN1 Snider Field

کلید یکتای فرودگاه IN1 است و نام فرودگاه Snider Field است.

❖ توجه کنید کد شما مورد بازبینی قرار میگیرد و باید از الگوریتم های گفته شده استفاده کنید. برای مثال برای بخش ۴ ممکن است Dijkstra هم جواب بدهد ولی پس از بازبینی اگر با الگوریتم موارد قبل پیاده سازی شده باشد نمره ۰ تعلق میگیرد.

❖ برای بخش ۴ علاوه بر پیاده سازی الگوریتم باید توضیحاتی درباره چگونگی انجام آن ارائه دهید. برای ارائه یک فایل حاوی توضیحات را در کوئرا آپلود کنید.

بخش اول: پیدا کردن تعداد مولفه های گراف

در این بخش هدف پیدا کردن تعداد دسته فرودگاه هایی است که به یکدیگر متصل اند. در واقع شما باید تعداد مولفه (component) های گراف ساخته شده را پیدا کنید.

خط اول ورودی شامل دو عدد n و m است که n تعداد فرودگاه ها و m تعداد یال هاست. هر یک از n خط بعد شامل اطلاعات هر فرودگاه که key و نام فرودگاه است میشود. در m خط بعد، هر خط شامل دو key است که نشان میدهد بین دو فرودگاه میتواند پرواز انجام شود (یال بدون جهت).

در تنها خط خروجی تعداد مولفه ها را چاپ کنید.

ورودی نمونه :

4 2

0MU8 Prince STOLport

31NH Johnson's Heliport

0MD4 Pond View Private Airport

0TA4 Erco Field

0MU8 31NH

0TA4

خروجی نمونه:

2

بخش دوم : پیدا کردن کمترین تعداد پرواز

در این بخش هدف پیدا کردن کمترین تعداد پرواز از فرودگاه مبدا به کلیه فرودگاه های دیگر است. اگر نتوان از مبدا به فرودگاهی سفر کرد برنامه باید مقدار ۱- را برای آن بازگرداند. خط اول ورودی شامل دو عدد n و m است که n تعداد فرودگاه ها و m تعداد یال هاست. هر یک از n خط بعد شامل اطلاعات هر فرودگاه که key و نام فرودگاه است میشود. در m خط بعد، هر خط شامل دو key است که نشان میدهد بین دو فرودگاه میتواند پرواز انجام شود (یال بدون جهت). خط آخر حاوی key فرودگاه مبدا است.

خروجی نام همه فرودگاه ها (به ترتیب ورودی) همراه با کمترین تعداد پرواز از مبدا به آن هاست. ورودی نمونه :

4 2

0MU8 Prince STOLport

31NH Johnson's Heliport

0MD4 Pond View Private Airport

0TA4 Erco Field

0MU8 31NH

0TA4 0MD4

0TA4

خروجی نمونه:

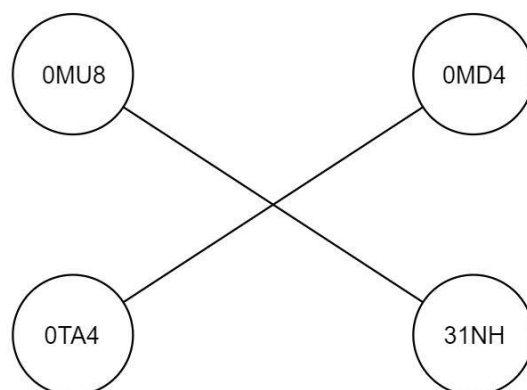
Prince STOLport -1

Johnson's Heliport -1

Pond View Private Airport 1

Erco Field 0

گراف حاصل به شکل زیر است و کمترین تعداد پرواز از مبدا 0TA4 را میخواهیم.



بخش سوم: پیدا کردن مسیر با کمترین هزینه.

در این بخش باید مسیری با کمترین هزینه را بین فرودگاه مبدا و مقصد پیدا کنید. برای انجام این بخش باید الگوریتم Dijkstra را پیاده سازی کنید.

خط اول ورودی شامل دو عدد n و m است که n تعداد فرودگاه ها و m تعداد یال ها است. هر یک از n خط بعد شامل اطلاعات هر فرودگاه که key و نام فرودگاه است میشود. در m خط بعد، هر خط شامل دو key و یک d عدد است که نشان دهنده پروازی دوطرفه (یالی بدون جهت) با قیمت بلیط d بین دو فرودگاه است. خروجی باید هزینه سفر (با دقت 2 رقم اعشار) به همراه key فرودگاه های درون مسیر باشد. اگر مسیری وجود نداشت 1- چاپ شود.

ورودی نمونه:

5 4

ABD Abadan

ACZ Zabol

ACP Maragheh Sahand

AAH Aachen Merzbrueck

AAK Aranuka

AAK ACP 136.81

AAH ACZ 12.74

ACP ACZ 14.63

ACZ ABD 60.48

ABD AAH 45.63

ABD AAK 514.74

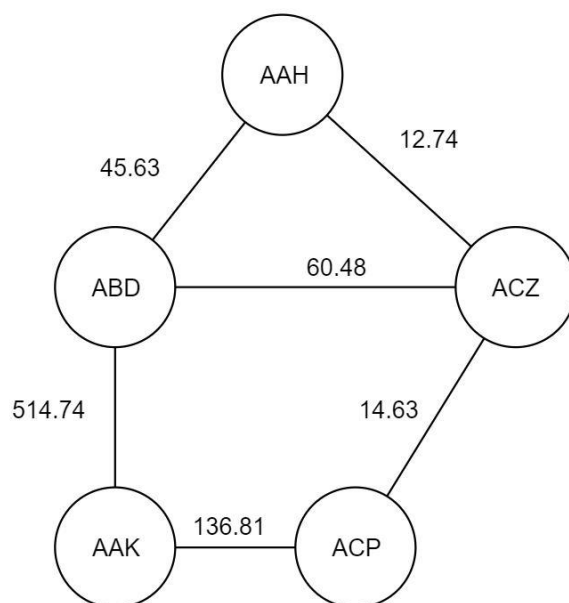
ABD

ACP

خروجی نمونه:

73.00

ABD AAH ACZ ACP



بخش چهارم: پیاده سازی بهینه تر

مشابه بخش قبل باید مسیر بین فرودگاه مبدا و مقصد را با کمترین هزینه پیدا کنید، در این بخش باید به دنبال الگوریتمی بهینه تر از Dijkstra باشیم. هزینه روی هر یال معادل قیمت بلیط پرواز بین دو فرودگاه است.

راهنمایی: میتوانید به عنوان یک جهت دهنده حرکت در گراف، فاصله مسقیم هر راس تا مقصد را در طول مسیر در نظر بگیرید. (قیمت بلیط با فاصله دو فرودگاه رابطه دارد)

خط اول ورودی شامل دو عدد n و m است که n تعداد فرودگاه ها m تعداد یال ها است.

هر یک از n خط بعد شامل اطلاعات هر فرودگاه که key ، مختصات و نام فرودگاه است میشود.

در m خط بعد، هر خط شامل دو key و یک d عدد است که نشان دهنده پروازی دوطرفه (یال بی جهت) با قیمت بلیط d بین دو فرودگاه است است.

دو خط آخر حاوی key فرودگاه مبدا و مقصد است است.

خروجی باید هزینه سفر (با دقت ۲ رقم اعشار) به همراه key فرودگاه های درون مسیر باشد. اگر مسیری وجود نداشت ۱- چاپ شود

ورودی نمونه

6 6

5 4

ABD 30.3711 48.2283 Abadan

ACZ 31.0983 61.5439 Zabol

ACP 37.3481 46.1331 Maragheh Sahand

AAH 50.8231 6.1864 Aachen Merzbrueck

AAK 0.1853 173.6369 Aranuka

AAK ACP 136.81

AAH ACZ 12.74

ACP ACZ 14.63

ACZ ABD 60.48

ABD AAH 45.63

ABD AAK 514.74

ABD

ACP

خروجی نمونه

73.00

ABD AAH ACZ ACP