

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT II BÀI TẬP LỚN 2

HỆ THỐNG QUẢN LÝ GIAO HÀNG

v 3.0

1. Giới thiệu

Một công ty chuyên sản xuất và cung ứng các mặt hàng ra thị trường. Công ty cần phải quản lý các quá trình giao hàng cho khách hàng.

2. Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập bản đồ khu vực giao hàng. Nhân viên giao hàng sẽ phải lựa chọn đường đi để giao hàng. Công ty cần tính chi phí giao hàng / khảo sát khu vực đối với từng loại nhân viên trong công ty. Chi tiết cụ thể công việc sinh viên phải làm sẽ mô tả trong phần 4.

3. Dữ liệu

Dữ liệu bản đồ khu vực giao hàng được lưu trong file mang tên map.txt. File này chứa thông tin về các đoạn đường có trong khu vực, các đoạn đường cách nhau bởi khoảng trắng hoặc dấu xuống dòng.

Mỗi đoạn đường được biểu diễn dưới dạng NNXXXMM, có nghĩa là có đường đi từ góc phố NN đến góc phố MM của khu vực, tuy nhiên chi phí để di chuyển trên đoạn đường này là XXX. Góc phố NN được gọi là góc phố khởi hành, góc phố MM được gọi là góc phố đích đến. Nếu nhân viên khởi hành vào giờ thấp điểm, có thể tự do đi về hai chiều giữa các góc phố. Tuy nhiên nếu giờ khởi hành là giờ cao điểm, nhân viên chỉ có thể đi từ góc phố khởi hành đến góc phố đích đến, không thể đi theo chiều ngược lại.

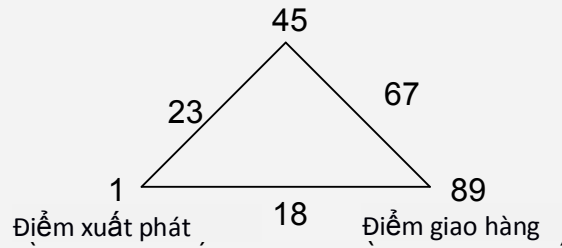
Số đoạn đường là không cố định, có thể thay đổi tùy theo test case. Giữa hai góc phố có thể có nhiều đoạn đường khác nhau. Sẽ có tối đa 1000000 đoạn đường.

Điểm xuất phát sẽ nằm ở góc phố khởi hành của đoạn đường đầu tiên, điểm giao hàng nằm ở góc phố đích đến ở đoạn đường cuối cùng trong file.

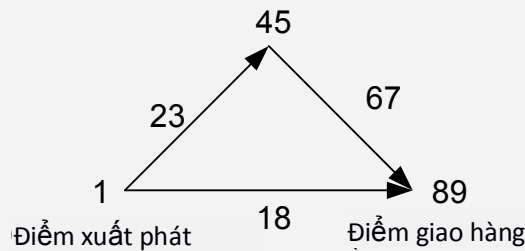
Ví dụ 1: Với dữ liệu bản đồ là

0102345 4506789 0101889

Nếu giờ khởi hành là giờ thấp điểm, bản đồ khu vực sẽ được biểu diễn như hình vẽ sau:



Nếu giờ khởi hành là giờ cao điểm, bản đồ khu vực sẽ được biểu diễn như hình vẽ sau:



4. Hiện thực

Sinh viên được yêu cầu hiện thực các phương thức sau trong lớp DeliveringMap:

- **public** DeliveringMap(File file): phương thức khởi tạo bản đồ khu vực từ dữ liệu được lưu trữ trong file;
- **public int** calculate(**int** level, **boolean** rushHour): phương thức tính chi phí giao hàng với 2 tham số là cấp độ của nhân viên giao hàng (level) và giờ khởi hành có phải là giờ cao điểm hay không (rushHour, true nếu giờ khởi hành là cao điểm); Cách tính chi phí được mô tả trong phần 5.

5. Cách tính chi phí giao hàng

C1) Nếu cấp độ (level) của nhân viên là 0 hoặc 1, đây là một nhân viên tập sự đi thám sát địa hình. Nhân viên tập sự này sẽ báo cáo lại thông tin khu vực dưới dạng một số nguyên I như sau:

$$I = \text{level} + N_p + \sum P_i$$

trong đó **level** là cấp độ của nhân viên, **N_p** là số các góc phố trong khu vực, **$\sum P_i$** là tổng chi phí các đoạn đường trong khu vực.

Ví dụ 2: Với dữ liệu bản đồ là

0102345 4506789 0101889

thì kết quả trả về của `calculate(1, true)`

$$1 + 3 + (23+67+18) = 112.$$

C2) Nếu L của nhân viên là 2, 3 hoặc 4, nhân viên sẽ chọn đường đi **có trọng số nhỏ nhất** từ Điểm xuất phát đến Điểm giao hàng. Tuy nhiên, nếu trọng số này lớn hơn $100 \cdot \text{level}$ của nhân viên giao hàng, nhân viên giao hàng sẽ không đủ sức khỏe đi hết quãng đường. Trong trường hợp đó, hàm trả về một số âm là hiệu số của $50 \cdot \text{level}$ và trọng số tìm được.

Trọng số của một con đường là tổng trọng số của tất cả các đoạn đường trên con đường đó.

Ví dụ 3: Với dữ liệu bản đồ là

0102345 4506789 0101889

thì kết quả trả về của `calculate(3, false)` là 18.

Ví dụ 4: Với dữ liệu bản đồ là

0102345 4506789 0111889

thì kết quả trả về của `calculate(3, true)` là $23+67 = 90$.

Ví dụ 5: Với dữ liệu bản đồ là

0109845 4509889 8919898

thì kết quả trả về của `calculate(3, false)` là $50*3-394 = -244$ vì $98+98+198 = 394 > 100*3$

C3) Nếu L của nhân viên giao hàng là 5 hoặc 6, nhân viên giao hàng này sẽ chọn con đường **có trọng số cao nhất** từ Điểm xuất phát đến Điểm giao hàng. Kết quả trả về sẽ là trọng số của con đường được chọn. Nếu con đường được chọn có trọng số cao hơn hoặc bằng $100*L$ của nhân viên, nhân viên này cũng sẽ không thể đi hết con đường và kết quả trả về là số âm của L của nhân viên.

Ví dụ 6: Với dữ liệu bản đồ là

0109845 4509898 0117298

thì kết quả `calculate(6, false)`

$98+98 = 196$.

Ví dụ 7: Với dữ liệu bản đồ là

0109845 4509898 0177298

thì kết quả trả về `calculate(5, false)` là -5.

Lý do là đường đi có trọng số cao nhất là $01 \Rightarrow 98$, trọng số $772 > 100*5$.

Chú ý là mặc dù còn con đường $01 \Rightarrow 45 \Rightarrow 98$ có trọng số 196 (nhỏ hơn $100*L$ của nhân viên giao hàng), nhưng con đường trọng số cao nhất là $01 \Rightarrow 98$ (trọng số 772) vẫn được chọn (và dẫn đến là nhân viên giao hàng đã bỏ cuộc trên đường đi).

C4) Nếu mã đoạn đường đi có dạng XX000YY, thì đoạn đường cho biết là đường đi từ đỉnh XX đến đỉnh YY có một cửa hàng giải khát. Các nhân viên giao hàng đều mong muốn đi ngang qua đó để giải lao. Con đường được chọn vẫn phải thỏa mô tả như đã mô tả trong mục C2 và C3. Nếu không tồn tại đường đi vừa giải khát vừa giao được hàng thì các nhân viên đành phải đi giao hàng trước. Kết quả trả về trong trường hợp như vậy sẽ giống như trong các trường hợp C2 và C3 đã nói. Chú ý: trong 1 khu phố, có tối đa 1 cửa hàng giải khát.

Ví dụ 8: Với dữ liệu bản đồ là

0111145 0119846 4699845 0100002 0211103 0312145 0210145

Kết quả trả về `calculate(6, false)` sẽ là $0+111+121 = 232$.

Có bốn con đường dẫn từ Điểm xuất phát đến Điểm giao hàng là $01 \Rightarrow 45$, $01 \Rightarrow 46 \Rightarrow 45$, $01 \Rightarrow 02 \Rightarrow 03 \Rightarrow 45$, và $01 \Rightarrow 02 \Rightarrow 45$. Trong đó chỉ có 2 đường cuối là đi ngang qua cửa hàng giải khát là $01 \Rightarrow 02 \Rightarrow 03 \Rightarrow 45$ có trọng số là $(0+111+121)$ và $01 \Rightarrow 02 \Rightarrow 45$ có trọng số là $(0+101)$. Tuy nhiên do nhân viên giao hàng có $level=6$, nhân viên này sẽ chọn con đường có trọng số cao hơn, kết quả trả về sẽ là $0+111+121 = 232$.

Ví dụ 9: Với dữ liệu bản đồ là

0111145 0119846 4699845 0100002 0211103 0399845 0299845

Kết quả trả về `calculate(5, false)` sẽ là -5.

Tương tự ví dụ 8, có 2 con đường có thể đi qua cửa hàng giải khát. Cả 2 đường này đều có trọng số lớn hơn $100*level$ của nhân viên giao hàng. Kết quả trả về là -5.

Ví dụ 10: Với dữ liệu bản đồ là

0111145 0119846 0100002 0211103 4699845

Kết quả trả về `calculate(3, false)` sẽ là 111.

Có hai con đường dẫn từ Điểm xuất phát đến Điểm giao hàng là $01 \Rightarrow 45$ và $01 \Rightarrow 46 \Rightarrow 45$. Trong đó không có con đường nào đi qua cửa hàng giải khát ở đoạn đường $01 \Rightarrow 02$. Do vậy nhân viên giao hàng này (có $level=3$) sẽ giao hàng như bình thường là chọn con đường có trọng số nhỏ nhất $01 \Rightarrow 45$.

C5) Nếu trọng số của đoạn đường là 99, đoạn đường này có nhiều lô cốt. Các nhân viên giao hàng đều ngại không muốn đi ngang qua đây và sẽ từ chối giao hàng nếu đường đi được chọn đi qua đoạn có lô cốt này. Trong trường hợp như vậy, kết quả trả về là $99 - 70 * level$ của nhân viên giao hàng.

Nếu đường đi được chọn không đi qua đoạn có lô cốt, kết quả là bình thường như trong C2 và C3. Chú ý: khi có lô cốt xuất hiện trong khu vực thì không có cửa hàng giải khát nào tồn tại cả.

Ví dụ 11: Với dữ liệu bản đồ là

0111102 0200145 0119945

Kết quả trả về `calculate(3, false)` là 112.

Ví dụ 12: Với dữ liệu bản đồ là

0111102 0200145 0109945

Kết quả trả về `calculate(3, false)` là $(99 - 70 * 3) = -111$.

C6) Nếu L của nhân viên giao hàng là 7, nhân viên này sẽ tìm đường đi có trọng số nhỏ nhất và đường đi có trọng số lớn nhất, sau đó tính trung bình cộng trọng số 2 con đường đã chọn. Nếu số trung bình cộng này mà nhỏ hơn $30 * level$ của nhân viên thì trả về kết quả này, ngược lại trả về -21.

Ví dụ 13: Với dữ liệu bản đồ là

0111102 0200145 0119945

Kết quả trả về `calculate(7, false)` là $(112 + 199) / 2 = 155$.

Ví dụ 14: Với dữ liệu bản đồ là

0111102 0200145 0139945

Kết quả trả về `calculate(7, false)` là -21 vì trung bình trọng số là $(112 + 399) / 2 = 255 > 30 * 7$

C7) Nếu L của nhân viên giao hàng là 9, nhân viên giao hàng đó chính là trưởng bộ phận. Nhân viên này muốn thám hiểm tất cả các góc phố ở khu vực. Kết quả trả về trong trường hợp này là tổng trọng số trên tất cả các cạnh của cây phủ tối thiểu đồ thị (minimum spanning tree) các đoạn đường ở khu vực giao hàng (tham khảo trong slide bài giảng hoặc tại địa chỉ http://en.wikipedia.org/wiki/Minimum_spanning_tree). Nếu không có tồn tại cây phủ tối thiểu nào, kết quả trả về là -3.

Chú ý: Trưởng bộ phận chỉ có thể làm được điều này khi đang trong giờ thấp điểm. Ở giờ cao điểm, nhân viên này không thể tìm ra cây phủ tối thiểu nào. (Kết quả sẽ trả về là -3.)

Ví dụ 15: Với dữ liệu bản đồ là

0111102 0200145 0109945 4511246 0100146

Kết quả trả về `calculate(9, false)` là $099 + 001 + 001 = 101$.

6. Cách dịch và thực thi chương trình

Sinh viên được cung cấp file assignment2.zip gồm các file sau:

- assignment2.pdf - file mô tả Bài tập lớn

- src\vn\edu\tdt\it\dsa\DeliveringMap.java - lớp bản đồ
- test\map.txt - ví dụ file lưu trữ bản đồ

Lưu ý:

- sinh viên viết mã cho các phương thức cần thực hiện được mô tả trong phần 4;
- được phép thêm các phương thức riêng hoặc thuộc tính riêng (private method, private field);
- không được phép thay đổi khuôn mẫu (prototype) các phương thức đã được định nghĩa;
- không được phép thay đổi tên package;
- file bài làm phải biên dịch được bằng Java phiên bản 1.8 và không sử dụng bất kỳ thư viện ngoài nào;
- comment phải viết bằng tiếng Anh hoặc tiếng Việt không dấu;

Lệnh biên dịch (trên Linux)

```
javac -d ./bin/ -sourcepath ./src/ ./src/vn/edu/tdt/it/dsa/DeliveringMap.java
```

Lệnh thực thi (trên Linux)

```
java -classpath ./bin/ vn.edu.tdt.it.dsa.DeliveringMap
```

7. Nộp bài

Khi nộp bài, sinh viên sử dụng account đã được cấp phát trên hệ thống Sakai để nộp bài qua mạng. Sinh viên chỉ nộp đúng một file **DeliveringMap.java**. Tất cả các file nộp khác sẽ bị tự động xóa khi chấm bài. File được nộp phải là file chương trình gốc, sinh viên không được nén file khi nộp bài. Thời hạn để nộp bài sẽ được thông báo trên Sakai. Sinh viên phải dùng account trên hệ thống Sakai để nộp bài. **KHÔNG** nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác. Bài nộp trễ sẽ **KHÔNG** được nhận.

8. Xử lý gian lận

Bài tập lớn phải được sinh viên **TỰ LÀM**. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, **TẤT CẢ** các bài nộp đều bị coi là gian lận. Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình. Các bài làm của các sinh viên ở các học kỳ trước cũng sẽ được dùng để kiểm tra gian lận.
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn. Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là **KHÔNG ĐƯỢC** sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học.

**KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO
KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!**

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.