CS2010 PS5 - The Onset of Labor

Released: Tuesday, 04 October 2011
Due: Saturday, 15 October 2011, 8am

Collaboration Policy. You are encouraged to work with other students on solving this problem set. However, you **must** write up your solution **by yourself**. In addition, when you write up your solution, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). Any deviation from this policy will be considered cheating, and will be punished severely, including referral to the NUS Board of Discipline. It is not worth it to cheat just to get 15% when you will lose out in the other 85%.

The Story. After ≈ 40 weeks, ≈ 9 months +, and 3 trimesters, it is now the time for birth... Jane's EDD (Expected Date of Delivery) is 22 October 2011 (transition of NUS Week10 to Week11). But as many parents will testify, babies rarely born *precisely* on their EDD, but they can be born +/- 2 weeks from the EDD. That's why, Steven, your CS2010 lecturer, must have a backup plan with his CS2010 lectures and tutorials around late October 2011.

For lectures, the possible actions are:

1. Cancelation of one of CS2010 lecture that falls on the same week as Jane's birth – a bit unlikely though, Steven does not want to take out 2 hours worth of lecture materials away from his students.
2. Another lecturer steps in during that 'labor week'.
3. Moving the affected lecture to another make up time slot.

After considering these various options, Steven finally choose to utilize recess week to do one normal lecture (option 3), effectively shifting forward Lecture 6/7/8/9 one week earlier, so that our schedule can be flexible on Week10/11/12. Mr (soon will be Dr) Chong Ket Fah (option 2) will then run a revision session during the affected lecture slot. For tutorials, Ket Fah will also take over Steven's classes (Tutorial Group 03 and 08) during the affected week.

When you first read this PS (early October 2011), Jane is probably just one or two weeks away from birth. By the time you have to submit this PS (Saturday, 15 October 2011), Jane is either already born or is about to be born... By solving this PS, you have a chance to help your lecturer do something super important in his life: Accompanying his wife during *the onset of labor*[1], escorting her from home to our chosen hospital as fast as possible, so that she can go through the 'challenging' three stages of labor[2] in a proper delivery suite in the hospital.

[1] The cues that birth is imminent, like the ruptured water bag, contractions, etc.
For more details, see: `http://www.babycenter.com/0_signs-of-labor_181.bc`

[2] See PS2 story: 'Scheduling Deliveries'. Now imagine that Steven's wife, Grace, is now really one of the women in the delivery room.

The Actual Problem. Given a map of Singapore (as a connected directed weighted graph), estimated time[3] to travel through Singapore roads (as *positive* weights of the corresponding directed edges – in minutes), Steven and Grace's home (vertex 0/'A' on that graph), their chosen hospital (vertex 1/'B' on that graph), determine the *quickest way* to go from Steven and Grace's home to their chosen hospital (the sum of edge weights along the shortest path). Steven will call for taxi and then instruct the taxi driver to take this path.

Steven and Grace needs the quickest way, because once the contractions start (**the onset of labor**), Grace does not want to spend *too much time* on the road because – as you may have guessed it correctly – it is 'not without pain'... Similarly, Steven does not want to be *overly anxious* and wants to have Grace taken care by the doctor and nurses/midwives *as soon as possible.* The first stage of labor usually last for *few hours.*

Now, let's go back to the problem. For example, suppose Singapore[4] a connected directed weighted graph as shown below (all edges are directed, except edge E-G is bidirectional, or we can view it as having two directed edges E->G and G->E):

```
     8    4         5
 A---->C<-------H---->B
 |     |        ^     ^
3|    4|       7|    2|
 v     v        |     |
 D---->E------->I     |
 |   6 ^     3  |     |
1|    1|       1|     |
 v     v        v     |
 F---->G        J---->K
     1              3
```

If Steven and Grace's home is at vertex 0/'A' and their chosen hospital is at vertex 1/'B', then the quickest way is this path: `A` → `D` → `F` → `G` → `E` → `I` → `J` → `K` → `B` with total estimated traveling time of: 3+1+1+1+3+1+3+2 = 15 minutes.

The skeleton program `Labor.java` is already written for you, you just need to implement one (or more) method(s)/function(s):

- `int Query()`
  Query your Adjacency List data structure[5] where the weight of each road (edge) is stored in the Adjacency List itself, return the shortest path weight from vertex 0 to vertex 1.
- If needed, you can write additional helper methods/functions to simplify your code.

Subtask 1 (25 points). On an 'impossible case' that simplifies this problem, the road network in Singapore is a tree and $1 \leq V + E \leq 100$.

```
     8              5
 A---->C        G---->B
 |              ^     ^
3|             7|
 v              |
 D---->E------->F
     6      2
```

The quickest way for the tree above is path: `A` → `D` → `E` → `F` → `G` → `B` with total estimated traveling time of: 3+6+2+7+5 = 23 minutes.

[3] To simplify this problem, let's assume that this time estimation is accurate and there is no traffic jam in Singapore.

[4] Yes, Singapore map does not looks like this, but let's just assume it is.

[5] Already implemented in `Labor.java`.

Subtask 2 (25 points). On another 'impossible case' that simplifies this problem, all roads in Singapore is a connected directed weighted graph but somehow require exactly 7 minutes to traverse and $1 \leq V + E \leq 100$.

```
     7     7
 A---->C---->B
 |     |     ^
7|     |    7|
 v     V     |
 D---->E---->F
     7       7
```

The quickest way for the graph above is path: `A` → `C` → `B` with total estimated traveling time of: 7+7 = 14 minutes.

Subtask 3 (25 points). The road network in Singapore is a connected directed weighted graph, the time to traverse Singapore roads varies, and $1 \leq V \times E \leq 1000000$. In fact, the sample test case shown above fits this description.

Subtask 4 (25 points). Same as Subtask 3, but now $1 \leq V + E \leq 250000$.

Note: The test data and the answers to reach 50 points: `Subtask1.txt`, `Subtask1-ans.txt`, `Subtask2.txt`, and `Subtask2-ans.txt` are given to you. The `Sample.txt` and `Sample-ans.txt` that contains *a small subset* of our official test data for Subtask 3 and 4 are also given. You are of course allowed and encouraged to generate larger test data (for Subtask 3 and 4) and cross check your program's output with your friend's on these additional test data. However, there is no bonus point for this PS.