CS2010 PS4 - Out For a Walk

Released: Thursday, 22 September 2011
Due: Tuesday, 04 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. By the time CS2010 students attempt this PS, Steven's wife (Grace) is already on her latest trimester (to be precise, Jane's birth is just a few weeks away). As you know, pregnant women are given priority seats in MRT (the corner seats at each car), buses (usually the front seats), and virtually at every other public places. We feel grateful every time a person who occupied one of those priority seats gave his/her seat to Grace (although we are also often irritated by young men[1] who 'ignored' the presence of a pregnant woman – usually either 'sleeping' or 'playing with his smart phone' – and did not give up their seats[2]).

Now, do you ever wonder why pregnant women are given priority seats (together with the disabled - obvious, the senior citizen - also obvious). This is because when a woman is pregnant, she becomes tired more easily. Since she is carrying a precious small human being in her womb, she must be guarded from all potential hazards, including the risks of falling (and miscarriage) because she cannot stand for too long in a crowded MRT or bus.

If you want to convince yourself that a pregnant woman will get tired easily when carrying a small baby in her womb, just carry a 10 kg rice pack for 10 minutes all around NUS.

Not just about MRTs and buses, pregnant women also need to take safer and easier paths when walking. Climbing a staircase requires a huge effort for pregnant women, so if there is a lift, an elevator, or a gradually increasing slope somewhere in that building, she will prefer to take the *easier path*, even if it means a longer path.

Grace wants to go out for a walk. Steven, as a Computer Scientist, wants to compute the best path for her :). But he 'needs' help from his CS2010 students.

[1] It is not surprising that usually the ladies who are more aware that Grace is pregnant and give up her seats.
[2] CS2010 students and teaching staffs!!, give up your seat in public places to those who need it more!!

The Actual Problem. Given a layout of a building (as a connected graph of course), Grace's effort rating to traverse the corridors of that building (as weights of the corresponding edges: lower weight means easier corridor for Grace, higher weight means harder corridor for Grace), Grace's starting point (a vertex on that graph), Grace's target point (another vertex on that graph), determine the maximum effort that Grace has to endure in order for her to go from the starting point to the target point (the edge with maximum weight along Grace's best/easiest path). Grace is not in rush. She can take a longer path (detour, etc) as long as her maximum effort that she has to endure along that path is minimized.

For example, suppose the building is a connected weighted graph as shown below:

```
   1       5
 A---B   E---F
3|  4|  7|  2|
 D---C---H---G
   6   3   1
```

If Grace wants to go from point `D` to point `F`, she will choose this path: `D` → `A` → `B` → `C` → `H` → `G` → `F`. This is not the shortest path, but it is the best/easiest path for her as she only needs to endure maximum effort rating of 4 when she goes through corridor `B-C`. The other corridors have effort ratings ≤ 4.

If Grace choose the shortest path (in terms of number of edges traversed): `D` → `C` → `H` → `G` → `F`, she has to endure a tougher corridor `D-C` (with an effort rating of 6) compared to her best path above.

The skeleton program `OutForAWalk.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[3] where the weight of each corridor (edge) is stored in the Adjacency List itself[4] and return the weight of a corridor (an edge) which has the highest effort rating along Grace's best path.
- If needed, you can write additional helper methods/functions to simplify your code.

Subtask 1 (50 points). The given building is a small weighted tree ($1 \leq V \leq 10$).

Example (Suppose Grace also wants to walk from point `D` to point `F`):

```
   1       5
 A---B   E---F
    4|  7|
 D---C---H---G
   6   3   1
```

In this building (weighted tree), Grace's best path is `D` → `C` → `H` → `E` → `F`. The hardest corridor (edge) for Grace is `H-E` with weight 7. The answer is 7.

Subtask 2 (20 points). The given building is a small weighted general graph ($1 \leq V \leq 10$).

Subtask 3 (20 points). The given building is a medium weighted general graph ($1 \leq V \leq 200$).

[3] Already implemented in `OutForAWalk.java` – by now you should be able to understand how it works.

[4] Unlike in PS3 where the weight of each room is stored in a separate Vector of Integers.

Subtask 4 (10 points). The given building is a large weighted general graph ($1 \leq V \leq 1000$). If you can solve this Subtask 4, you will get 50+20+20+10 = 100 points. There is no bonus point for this Subtask as the solution is not too difficult if you understand the lab demo during Week07.

Note: The test data to reach 70 points: `Subtask1.txt` and `Subtask2.txt` are given to you. You are allowed to check your program's output with your friend's.