CS2010 PS4 - Out For a Walk v2

Released: Thursday, 27 September 2012
Due: Tuesday, 09 October 2012, 8am

Collaboration Policy. You are encouraged to work with other students or teaching staffs (inside or outside this module) on solving this problem set. However, you **must** write the Java code **by yourself**. In addition, when you write your Java code, 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). This list may include certain posts from fellow students in CS2010 IVLE discussion forum. 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%.

R-option. There is no R-option in this PS. Time to cool down until PS7 =).

Last Year's Story (still relevant today). By the time (last year) CS2010 students attempted this PS, Steven's wife (Grace) was on her latest trimester (to be precise, Jane's birth was just a few weeks away from the deadline of PS4 last year). Today, Jane is already born but still very small. As you know, pregnant women and parents carrying small babies 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/parent carrying a small baby – usually either 'sleeping' or 'playing with his smart phone' – and do not give up their seats[2]).

Now, do you ever wonder why pregnant women/parents carrying small babies are given priority seats (together with the disabled - obvious, the senior citizen - also obvious). 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. Similarly for a parent carrying small baby.

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.

[1] It is not surprising that it is usually the ladies who are more aware of another woman's pregnancy and gives up her seats.

[2] CS2010 students and teaching staffs!!, give up your seat in public places to those who need it more!!

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.

Last year, when Grace was pregnant, she wanted to go out for a walk. Steven, as a Computer Scientist, wanted to compute the easiest path for her :). Of course Steven can code the solution by himself, but he have just taught 'something' to his CS2010 students that can be used to solve this problem and he gives his students a chance to help him help his wife (last year).

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 source vertex, Grace's destination vertex, determine the maximum effort that Grace has to endure in order for her to go from the source vertex to the destination vertex (the edge with maximum weight along Grace's 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. There will Q queries with varying source and destination vertices. The source vertices can only range from [**0..9**] while the destination vertices can range from [**0..V-1**]).

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

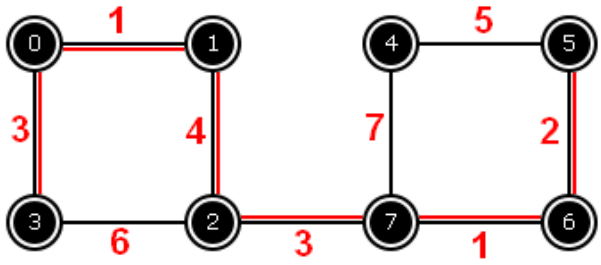

Figure 1: A Sample Building

If Grace wants to go from point `3` to point `5`, she will choose this path: `3` → `0` → `1` → `2` → `7` → `6` → `5`. This is *not* the shortest path, but it is the easiest path for her as she only needs to endure maximum effort rating of 4 when she goes through corridor `1-2`. The other corridors along this easiest path have effort ratings ≤ 4. If Grace choose the shortest path (in terms of number of edges traversed): `3` → `2` → `7` → `6` → `5`, she has to endure a tougher corridor `3-2` (with an effort rating of 6) compared to her easiest path above.

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

- `void PreProcess()`
  This is an optional method that you may choose to use to speed up your queries.
  You can leave this method blank if you do not need it.
- `int Query(int source, int destination)`
  Query your chosen data structure and return the weight of a corridor (an edge) which has the highest effort rating along Grace's easiest path from source to destination.
- If needed, you can write additional helper methods/functions to simplify your code.

Subtask 1 (20 points). The building is a small weighted tree ($1 \leq V \leq 10$, $1 \leq Q \leq 5$).

Example (source: `3`, destination: `5`): In this building (weighted tree), Grace's easiest path is `3` $\rightarrow$ `2` $\rightarrow$ `7` $\rightarrow$ `4` $\rightarrow$ `5`. The hardest corridor (edge) for Grace is `7-4` with weight `7`. The answer is `7`.

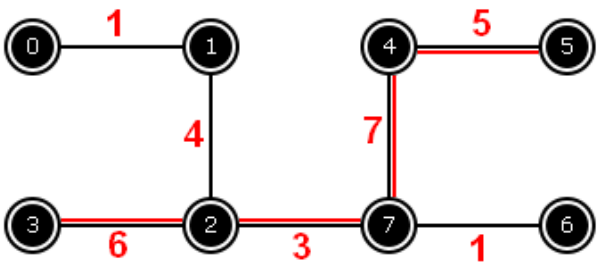

Figure 2: A Sample Building (Tree)

Subtask 2 (Additional 20 points). The building is a small weighted graph ($1 \leq V \leq 10$, $1 \leq Q \leq 5$).

Subtask 3 (Additional 20 points). The building is a medium weighted graph ($1 \leq V \leq 400$, $1 \leq Q \leq 5$).

Subtask 4 (Additional 20 points). The building is a large weighted graph ($1 \leq V \leq 2000$, $0 \leq E \leq 100000$, $1 \leq Q \leq 5$).

Subtask 5 (Additional 20 points). The building is a large weighted graph ($1 \leq V \leq 2000$, $0 \leq E \leq 100000$, $1 \leq Q \leq 100000$).

Note: The test data to reach 40 points: `Subtask1.txt` and `Subtask2.txt` are given to you. You are allowed to check your program's output with your friend's. You are encouraged to generate and post additional test data in IVLE discussion forum.