CS2010 PS6 - Caesarean Section v2

Released: Thursday, 18 October 2012 (actually Tuesday, 16 October 2012)
Due: Tuesday, 30 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. This was a scenario that myself (Steven) and my wife (Grace) initially hoped that we did *not* have to go through: a Caesarean section *surgery*. However, Grace eventually had to go through this surgery procedure and we are thankful for the inventor of this technique and current medical knowledge to make this surgery safer.

Caesarean section is an *alternative* method of delivering a baby. It is a **surgical** procedure that involves the obstetrician (the doctor) *'cutting'* through the mother's abdomen, take out the baby, and *suture* the mother back to her original state. Sounds scary? You will probably think so after looking at this video: `http://video.about.com/pregnancy/Cesarean-Section.htm`.

In Singapore, most doctor will by default choose *normal delivery*. Only if there are *known complications* prior to the delivery (e.g. the baby is in breech position, etc) or *unexpected complications* during the attempted normal delivery (e.g. fetal distress, umbilical cord prolapse, etc), then the doctor may offer to perform this alternative delivery method.

As Caesarean section is a surgical procedure, it must be performed *'as fast as possible'* (you do not want the mother bleeds to death or the baby dies). However, it has step-by-step procedures that must be performed one after another (e.g. you have to put the mother on anaesthetic first before cutting her – you do not want to reverse these two steps, etc). Each step has an estimated completion time. Sometimes, nurses, midwives, and/or paediatrician (child doctor) can help do some steps so that the overall surgery time can be minimized to reduce the father's anxiety.

For example, suppose an *over-simplified* Caesarean Section surgery is as follow:

0. The mother is put on anaesthetic, the baby is still in the womb (10 minutes)
1. After step 0, Doctor cuts the mother at the correct spots (15 minutes)
2. After step 1, Doctor takes out the baby carefully (3 minutes)
3. After step 2, Doctor sutures the mother back (30 minutes)
4. After step 2, Nurse cleans the newborn baby (10 minutes)
5. After step 4, Paediatrician measures the newborn baby's birth parameters (5 minutes)
6. After step 3 and 5, present the mother AND baby (both alive) to the father (0 minute)

These 7 steps can be visualized graphically as follows:

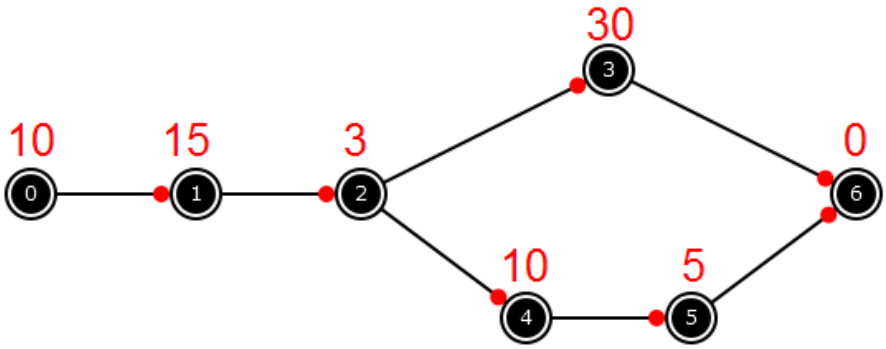

Figure 1: An Over-Simplified Caesarean Section Surgery

In the example above, the start of the surgery is step 0, when the father know the baby is still *inside* the mother's womb and see that the mother is put on anaesthetic. The end of the surgery is step 6 when the father see the living baby *outside* the womb **and** he also see his wife alive. Here, the father has to wait anxiously for 10+15+3+30 = 58 minutes to get **both** good news.

The Actual Problem. Given a dependency information between various steps of a Caesarean section surgery (as a directed graph that will not have a cycle), estimated time to perform such steps (as weights of the corresponding vertices – in minutes), determine the *quickest time* to complete the whole Caesarean section surgery, i.e. from the first step 0 (the mother is put on anaesthetic) until the last step V-1 (the father sees both the mother and the baby alive) can be completed.

Each step can only be performed by one person (usually the doctor). However, if there are two (or more) steps that can be performed by another qualified persons (another doctor, nurse, midwife, paediatrician, even the father, etc), the doctor *can always* call for enough number of helpers to speed up the process while the doctor is taking care on one other step.

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

- `int Query()`
  You are given an (unweighted) Edge List data structure[1] **EL** that stores the dependency information between various V steps of a Caesarean section surgery. An edge (u, v) in **EL** implies that step u must be performed *before* step v. You can assume that there is always a sequence of steps from step 0 that ends up at step V-1. You can also assume that only vertex 0 has 0-incoming degree in the given graph. You are also given an array **estT** of size V that stores the estimated time to complete each step (step V-1 always has 0 minute; the other steps have positive minutes). Query these two data structures and answer the query as defined above.

- If needed, you can write additional helper methods/functions to simplify your code.

Subtask 1 (25 points). In this subtask, the doctor already plan the sequence of steps very carefully so that by executing step 0, 1, 2, ..., V-1, in that order, he/she will be able to complete the surgery successfully without violating any step dependency[2]. The sample shown above (see Figure 1) fits this description. Constraints: $1 \leq V, E \leq 10$.

Subtask 2 (Additional 50 points). In this subtask, the doctor has *not* plan the sequence of steps yet. You have to help him/her plan the surgical steps (so that he/she can ask enough number of other person to help performing some of the steps). Constraints: $1 \leq V, E \leq 10$.

[1] Already implemented in `Caesarean.java`.
[2] Violation of this step dependency can harm the mother or the baby.

Subtask 3 (Additional 25 points). Same as Subtask 2, but $1 \leq V, E \leq 200000$. As the test data is large, you have to come up with the *most efficient* solution that you can think of.

Note: The test data to reach 75 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. All official test data has been verified correct using `CaesareanVerify.java` – however, as this checker program has similar features with the required solution, Steven decides not to release it. Student who wants to post additional test data in forum shall check for these conditions:

1. The last index of `estT` must be 0 whereas the other indices must be positive.
2. The graph must be a Directed Acyclic Graph.
3. Only vertex 0 has 0-incoming degree in the given graph.
4. There must be at least one path from index 0 to index V-1.