CS2010 PS2 - Scheduling Deliveries

Released: Thursday, 01 September 2011
Due: Saturday, 10 September 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. June 2011 is part of the long NUS 'University holiday' period this year. Although NUS teaching staffs are not actually on holiday[1], June 2011 was considered a lighter month for Steven. Therefore, Steven and his wife, Grace, decided to enroll in birth classes at a certain hospital in Singapore (name omitted to avoid indirect advertising).

During one of the session, we are escorted to do a 'hospital tour' (this will be discussed in more depth in the next PS3). One of the room type that is shown during the hospital tour is the 'delivery suites'. There are several rooms that are designed for women to deliver their babies. Those rooms have special delivery bed, pain relief systems: laughing gas 'Enthonox', epidural anaesthesia, and lots of other stuffs to make the mother-to-be as comfortable as possible.

Steven spotted a big white board in the reception desk and saw several pregnant women names but only one obstetrician[2]/gynaecologist[3] (let's just called him/her 'the doctor') on duty at that time. So Steven is quite sure that the doctor must be praying that not all pregnant women gave birth at the same time. In such situation, the doctor must decide which one of these women to give more attention based on their 'dilation status' (see below).

In the 'three stages of labor' (this will be discussed in more depth in PS5), the longest part is stage one: Dilation. Let's omit the detailed description and let's just say that dilation is measured in centimeters: Usually starts from about 3.0 cm (the start of labor) to around 10.0 cm (the baby is ready to be 'pushed' out/born). Anyway, if you are interested, please read this article: `http://en.wikipedia.org/wiki/Childbirth`.

[1]They have to prepare for the upcoming semester, or to continue with their research duties, or for Steven's case – prepare the Singapore International Olympiad in Informatics (IOI) team.

[2]Medical specialty dealing with the care of all women's reproductive tracts and their children during pregnancy (prenatal period), childbirth, and the postnatal period.

[3]Medical practice dealing with the health of the female reproductive system, or simply said "the science of women".

The Actual Problem. Given N pregnant women names that are about to give birth and their initial dilation status, determine which woman that the doctor on duty has to give his/her most attention to. A woman with higher dilation status has higher priority. If there are more than one woman with the same highest dilation status, the doctor will give priority to the woman who arrived at the hospital earlier.

The skeleton program `SchedulingDeliveries.java` is already written for you, you just need to implement four more methods/functions:

- `void ArriveAtHospital(String womanName, double dilationStatus)`
  Insert this `womanName` and her initial `dilationStatus` upon arrival at hospital into a suitable data structure of your choice. `womanName` is a String that contains only uppercase alphabets with length up to 100 characters. The women names are all unique.

- `void UpdateStatus(String womanName, double increaseDilation)`
  Medically, `dilationStatus` can only go up to around `dilationStatus = 10.0` centimeters. However, to simplify this problem, we will not bother with the maximum value. This method will simply update the dilation status of `womanName` from `dilationStatus` to `dilationStatus + increaseDilation` even if this causes `dilationStatus` $>$ `10.0`. You can safely assume that `ArriveAtHospital(womanName, any positive value` $\geq$ `3.0)` is already called prior to calling this method.

- `void GiveBirth(String womanName)`
  Medically, it takes several minutes or even hours from dilation around 10.0 centimeters until the baby is actually born. Some mothers can actually deliver the baby even if her dilation is still less than 10.0 centimeters (but not an ideal situation). Again, to simplify this problem, we assume that upon calling this method, the `womanName` gives birth in 'that instant' and no longer need to be taken care by the doctor.

- `String Query()`
  Query your data structure and reports the name of the woman that the doctor has to give the most attention to. See the priority criteria defined above. If there is no more woman to be taken care of, return a String: "The delivery suite is empty".

Example:
Let the sequence of events are as follows:

- `ArriveAtHospital(''GRACE'', 3.1)`

- `ArriveAtHospital(''ASTRID'', 5.5)`

- `ArriveAtHospital(''MARIA'', 4.23)`

- `Query()`
  You have to print out "ASTRID", as she is currently the one with highest `dilationStatus`.

- `ArriveAtHospital(''CINDY'', 7.77)`

- `Query()`
  Now you have to print out "CINDY".

- `UpdateStatus(''GRACE'', 2.4)`
  You still have to print out "CINDY" because "GRACE" now has `dilationStatus = 3.1 + 2.4 = 5.5`, still about 2 centimeters smaller than "CINDY". Note that "ASTRID" also has `dilationStatus = 5.5`.

- `GiveBirth(''CINDY'')`
  "CINDY" now gives birth 'instantly', and she is no longer in the doctor's radar.

- `Query()`
  Now you have to print out “GRACE”, because although “ASTRID” also has the same `dilationStatus = 5.5`, “GRACE” arrives at the hospital earlier than “ASTRID”, thus she is given higher priority.
- `GiveBirth(‘‘MARIA’’)`
  Suddenly “MARIA” reaches dilation status 10.0 and gives birth instantly.
- `Query()`
  The answer is still: “GRACE”.
- `GiveBirth(‘‘GRACE’’)`
- `Query()`
  You have to answer: “ASTRID”.
- `GiveBirth(‘‘ASTRID’’)`
- `Query()`
  “The delivery suite is empty”.

Subtask 1 (50 points). In this subtask, there is no call to `UpdateStatus` method. The method `GiveBirth` is always called for the woman currently under the doctor’s highest priority (you can view this as `GiveBirth(Query())`. For this simpler Subtask 1, you can assume that `dilationStatus` value when method `ArriveAtHospital(womanName, dilationStatus)` is called is always an integer between `[3..10]`. The number of women involved in this subtask is ≤ 10.

Subtask 2 (25 points). Same as Subtask 1, but `dilationStatus` is now a floating point number.

Subtask 3 (15 points). Unlike Subtask 1, this time you *also* have to deal with `updateStatus` method and women `GiveBirth` in *any* order as shown in the example above. However, the number of women involved in this subtask is still ≤ 10.

Bonus point: (20 points). Your program must be able to solve Subtask 3 and be very efficient in order to handle up to ≤ 10000 women arriving in the hospital and frequently updating their dilation status. Hint: All operations must be $O(\log n)$. If you can do this, your score will be $50+25+15+20 = 110$ points.

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.