CS213M: Assignment 4

Problem 2: A Discrete Event Simulation

Due Date: 13/03/2015

We want to simulate a culture of microbes in this problem. We start with an initial population of microbes and live (simulate) their lives for them. So we want to check what happens when these organisms undergo cell division and produce new ones, get infected, and when they die.

In order to do so, we use a kind of simulations called discrete event simulation. Read up about this on the internet if you are not already familiar with it.

Going into the gory details of the problem, there will be the following events in your simulation.

1. Birth: A microbe divides into two, the original parent lives on and a new one is born
2. Infection: A microbe gets infected
3. Death: A microbe dies, when it's lived it's lifetime or due to an infection

The constraints of the problem are as follows.

1. Each microbe has a positive integral '`strength`' associated with it.
2. This strength decides how many times a microbe can divide and how long it lives after it has been infected. The strength of a microbe decreases by one every time it divides, while the strength of its child will be equal to the strength of the parent when the parent was born.
3. If a microbe gets infected, it lives on for more (current) strength time steps. A microbe with less strength will thus die off faster. An infected microbe cannot undergo cell division.
4. A microbe can and will divide `n` time steps after its birth and this will also be the time before a microbe divides again after a cell division. A microbe dies `n` time steps after it undergoes it's last cell division. For each microbe, `n` will be a function of it's identity[1] and it's strength as follows.

   ```
   n = ((identity + strength) * 29) % 37
   ```

5. In a single time step, more than one events may occur concerning the same microbe. For example, a microbe might be scheduled to divide while there is an input event saying it gets infected too at the same time. We therefore assume the following priority of events.

   death > infection > division

6. If more than one microbes divide at the same time, you should assign identities to the new microbes in the same order as the order of their parents' identities.
7. Time starts from 0.

The inputs the problem will be an initial population of microbes to start off with given with their respective strengths, the parameter n, and instances of infection events. The expected output is birth and death of microbes and the size of the population at the end of the simulation. The exact format for input and output is described later.

An Example:

Let the initial population be of two microbes 0 and 1, with initial strengths 9 and 7 respectively. The values of n for them will thus be 2 and 10 respectively, also say microbes 0 and 3 get infected at

---

[1] We number microbes starting from 0 and keep incrementing the number. We assume the size of the population is always less than what fits in a 32 bit unsigned integer.

timesteps 4 and 11, and we have to simulate the culture till tilmestep 21. The simulation proceeds as follows then.

Timestep	Event
2	Birth of microbe 2 (microbe 0 divides)
4	Infection of microbe 0
10	Birth of microbe 3 (microbe 1 divides)
11	Infection of microbe 3
12	Death of microbe 0
18	Death of microbe 3
20	Birth of microbe 4 (microbe 1 divides)

Thus at the end, the size of the population is 3, with microbes 1, 2, and 4 alive.

Input Format:
The first line will contain N, the size of the initial population. N lines follow, each with the strengths of the microbes in the initial population starting from 1.
The next line contains M. M lines follow with the numbers of microbes getting infected along with the time step a microbe gets infected at, the two being separated by a space.
The next line contains T, the time step till which you have to simulate the culture before stopping and reporting the size of the population

Output Format:
You have to report births and deaths as and when a microbe is born or when it dies as follows.

`<event> <microbe identity> <timestep>`

event can be b or d representing a birth or a death respectively.
At the end, you have to output the size of the final population.

You have to submit a single program in a file named **`discreteSim.cpp.`**

Hint: In order to solve the problems of deleting events from the priority queue you use, use the heap implemented in problem 1. If you do that you would have to inherit from the `pred` class and define your own subclasses that have the `toDelete` function you need. Try adding member variables to the subclass and use them in the `toDelete` method to cleverly delete selective events on the fly.