

Data Structures and Introduction to Algorithms

Due: 10/2/2015 at noon (12:00pm)

1 Assignment 4: Priority Queues

A Scheduling Problem

You are in charge of writing a scheduler for a widget machine. Every 10 seconds, a new job comes in. A job has a deadline, and a size. The size is the number of *seconds* it will take to complete the job. Only one job can be in process at a time. Once the widget machine starts on a job, it must finish it. Some jobs are worth more money, a value. Every time a job finishes, the next job should be the one with the highest value, but no job should be started if it can't be finished in time for the deadline.

Write a priority queue to process the jobs and report which ones will finish.

Line i of the input contains the i th job that appears after i hours exactly. The job has 3 numbers: the value, the size, and the deadline (in that order). Here is a sample input:

```
12 35 35
65 25 50
30 30 300
40 50 70
2 3 100
```

This input has 5 jobs. Job 0 comes in at time 0. It has the highest value of any job seen thus far, so it gets started. It will take 35 seconds, so it will complete at time 35. The deadline is 35, so it will complete on time, and thus will be made. At time 10, job 1 arrived. At time 20, job 2 arrived. At time 30, job 3 arrived. At time 35, when job 0 finishes, the highest value job is job 1 (value 65). However, if it was started now, it would finish at time 60. This is after the deadline, so job 1 cannot be completed in time. The next highest value job is job 3 (value 40). It also cannot be completed in by the deadline. The next highest value job is job 2 (value 30). It has size 30. It will finish at time 65 which is less than its deadline. So, this will be the next job. At time 40, job 4 arrives. When job 2 finishes at time 65, job 4 is the highest priority job. It can be completed at time 68, which is less than the deadline, so it will be completed. The output will be the list of completed jobs in the order they are completed.

```
0 2 4
```

Time starts at 0 and is counted in seconds. This is when job 0 arrives. Assume that the scheduling takes a negligible amount of time.

2 Project Specification

Put your `main` method in a class called `PriorityQueueProject`. You should implement your own priority queue data structure. You will not receive full credit if your data structure does an exhaustive search for `insert` or `removeMax` (as in the list implementations). Think about heaps instead.

Input/Output Example 1

Input:

```
1 30 30
5 10 100
7 10 100
```

Output:

```
0 2 1
```

Input/Output Example 2

Input:

```
9 100 100
2 100 1000
4 100 1000
6 100 1000
8 100 1000
14 100 1000
5 100 1000
7 100 1000
3 100 1000
19 200 1000
13 200 1000
21 100 1000
```

Output:

```
0 9 11 5 10 4 7 3
```