

# CS6140: Advanced Programming Lab

## Assignment 2

Lead TA: Sudhir Samrit.  
August 7, 2014

### Description

Write a program to implement heapsort using pointer-based binary heaps.

Each node in heap contains three integers  $\langle jobId, priority, duration \rangle$ . Maintain heap property by comparing ‘priority’ values, i.e., each node in heap should have priority greater than or equal to priorities of its children (smallest value have highest priority). Use following rules for breaking ties:

- If nodes have same ‘priority’ values then use ‘duration’ field to break ties, i.e., node having smaller ‘duration’ should precede to node having larger duration.
- If nodes have same ‘priority’ as well as same ‘duration’ values then use ‘job id’ field to break ties, i.e., node having smaller ‘job id’ should precede to node having greater ‘job id’ (assume job id’s are distinct).

Read input from file then perform heapsort and write output into another file.

### Input

The first line of input file contains N, the number of jobs. Followed by N lines which contain three numbers jobId, priority and duration each separated by a space. Sample input file:

```
6
1 12 200
2 10 150
3 9 100
4 7 150
5 12 50
6 9 100
```

### Output

Output file should contain sorted entries in exactly same format as that of input file. Sample output file:

```
6
4 7 150
3 9 100
6 9 100
2 10 150
5 12 50
1 12 200
```

## Submission guidelines

- Your program should read input and output file names as command line arguments.  
For Example: `./a.out < InputFile > < OutputFile >`
- Whole source code should be in single file having name as `< roll_number.cpp >`.  
For Example: CS13M080.cpp
- Compress `< roll_number.cpp >` file into `< roll_number.cpp.tar.gz >` before uploading in moodle.