Due: 5th of September 2016 at 11:59pm

COMP 2007 – Assignment 2

All submitted work must be done individually without consulting someone elses solutions in accordance with the Universitys Academic Dishonesty and Plagiarism policies.

IMPORTANT! Questions 1a-c and 2a-c should be submitted via Blackboard as pdf (no handwriting!). The implementation required for Questions 1d and 2d should be done in Ed, and submitted via Ed.

Questions

Both questions in this week's assignment will be looking at the top hull problem. The *top hull* of a set S of n lines segments in the plane is an important concept for many applications, including visibility and motion planning. The segments are regarded as opaque obstacles, and their top hull consists of the portion of the segments visible from the point $(0, \infty)$. We will study the special case when all the segments in S are horizontal segments with y-coordinates greater than or equal to 0.

An example of the input is given in Fig. 1(left); the corresponding output is given in Fig. 1(right).

A horizontal segment s_i in S is represented by the triple (l_i, r_i, h_i) where l_i and r_i denote the left and right x-coordinates of the segment respectively, and h_i denotes y-coordinate of s_i. The input is a list of triples; one per segment. The output is the top hull specified as a list of horizontal segments arranged in order by their left x-coordinates. For the example shown in Fig. 1, the input and output are:

Input			Output		
0	3	3	0	2	3
2	4	4	2	4	4
1	5	2	4	5	2
6	7	4	5	6	3
5	8	3	6	7	4
9	11	3	7	8	3
9	12	1	9	11	3
			11	12	1

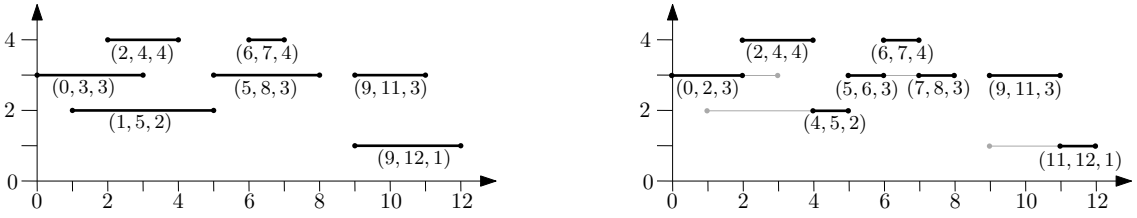

Figure 1: (left) Example of input. (right) Illustrating the output.

1. **[60 points]** We will solve the top hull problem using a divide-and-conquer approach. Recall from the lecture that a divide-and-conquer algorithm works by recursively breaking down a

problem into two or more subproblems of the same type, until these become simple enough to be solved directly. The solutions to the subproblems are then merged into a solution to the original problem. Your task for Question 1 is to handle the merge step. That is, as a subroutine to the divide-and-conquer algorithm (Question 2) you will need `MergeHulls`(H_1, H_2) that takes two top hulls as input and produces a single top hull H of them. An example is shown in Fig. 2, where two top hulls are given as input and "merged" into a single top hull.

You may assume that the two top hulls passed as arguments to `MergeHulls` are given as a list of horizontal segments ordered from left to right.

(a) Description of how your algorithm works ("in plain English"). For full points the algorithm should run in $O(n)$ time where n is the total number of segments in H_1 and H_2. [15 points]

(b) Argue why your algorithm is correct. [10 points]

(c) Prove an upper bound on the time complexity of your algorithm. [5 points]

(d) Implement the function `MergeHulls` (in Ed). [30 points]

An instance is given as two top hulls, where each hull is a list of horizontal segments ordered from left to right. Each horizontal segment s_i in a top hull is represented as a 3-tuple (l_i, r_i, h_i), where l_i is the x-coordinate of its left endpoint, r_i is the x-coordinate of its right endpoint, and h_i is its y-coordinate. The data files use the following format:

l_1 r_1 h_1
l_2 r_2 h_2
l_3 r_3 h_3
...

The segments along the top hull should be "maximal", that is, two segments $(0, 3, 1)$ and $(1, 4, 1)$ should be represented as $(0, 4, 1)$.

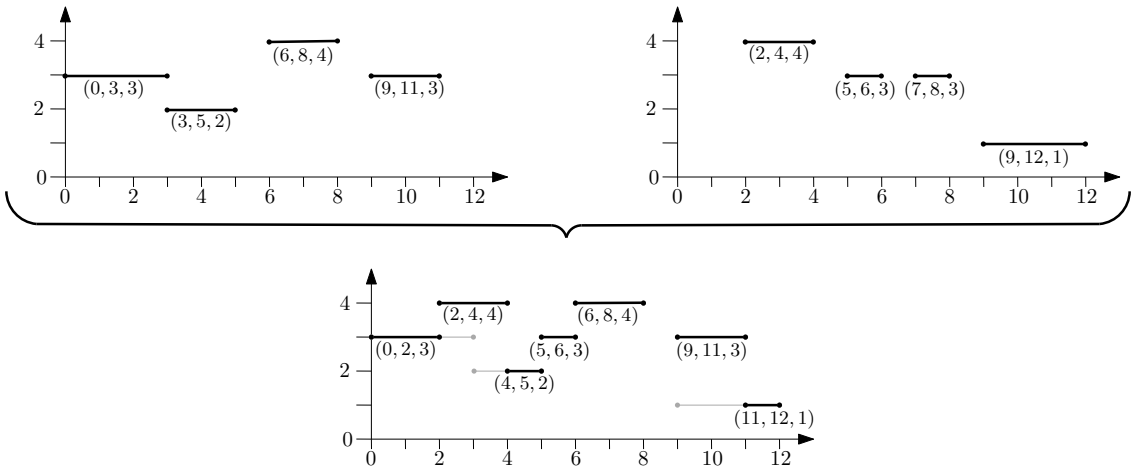

Figure 2: Illustrating `MergeHulls` that takes two top hulls of horizontal segments as input and "merged" them into a single top hull.

2. **[40 points]** Using the function `MergeHulls`(E_1, E_2) from Question 1 your task is to design a Divide-and-Conquer algorithm that computes the top hull of a set S of n horizontal line segments in the plane. Each segment $s_i \in S$ is represented (as above) as a triple (l_i, r_i, h_i). An example of the input instance is given in Fig. 1(left) and the top hull is highlighted in Fig. 1(right).

The output must follow the input format to `MergeHulls` stated in Question 1, that is, list the horizontal segments in the top hull from left to right. For example, the top hull shown Fig. 1(right) must be listed as:

```
0 2 3
2 4 4
4 5 2
5 6 3
6 7 4
7 8 3
9 11 3
11 12 1
```

(a) Description of how your algorithm works ("in plain English"). For full points (and assuming that `MergeHulls` runs in $O(n)$ time) the algorithm should run in $O(n \log n)$ time where n is the number of segments in S. [15 points]

(b) Argue why your algorithm is correct. [5 points]

(c) Prove an upper bound on the time complexity of your algorithm. [10 points]

(d) Implement your algorithm (in Ed). Note that this question requires that you already have a working implementation for Question 1d. [10 points]