

Assignment 2

Satyanand
14EC10049

Problem 1

This problem is popularly known as the skyline problem. A skyline can be understood as the union of two or more buildings in the 2-D plane. We can solve it using divide and conquer approach as the union can be taken in steps.

Algorithm

- We can first sort the buildings by their starting x position.
- Now we can divide the problem into two subproblems. Finding the union of $b_0, b_1 \dots b_n$ can be done by finding union of $b_0, b_1 \dots b_{n/2}$ and the rest and then taking union of the two skylines obtained.
- The process of merge is very similar to the mergesort technique. We sweep over two skylines and keep on adding the higher one first and discarding the lower one.
- We can extend this methodology recursively.
- The base case is when we have either only one building. In this situation the skyline consists of only that one building. So we return this building in skyline form.

Complexity Analysis

Here the number of levels of recursion(recursively calling union) is $\lg(n)$. At each level we are taking union by looping over the available buildings. Each building is traversed only once at each level hence the time complexity of each level is $O(n)$. Hence the overall complexity of the algorithm is $O(n \lg n)$.

Problem 2

- Now, when the rooftops of the buildings are slant the overall idea of solving remains the same. What changes here is the merge procedure of the two skylines. Here we have to keep in mind that the second skyline can become higher than the first one and vice-versa even when no building starts or ends at that particular x position. This is because the slant roof can keep on decreasing(or increasing) and intersect the other skyline(and hence the skyline higher from that instant changes).

Problem 3

To find the convex hull of a fully connected skyline we can loop over the available points.

We first push the first two points to the stack.

Now we loop over the available points in order of increasing x value. If adding the new point doesn't violate the convex property of the curve formed by points in the stack we add this point to the stack.

If it violates the property it means that the previous point is the point responsible as it lies below the current point and the point previous to it in the stack. So we pop the topmost point from the stack and again consider the point in question(whether it can be added to the convex hull stack)

Complexity Analysis

Since we loop over the points only once and each loop requires constant operations the time complexity is $O(n)$.