Report for assignment 3

Manne Naga Nithin (14EE10026)

January 22, 2016

A. Skyline of a city with flat rooftops

1. **First step** We take the input of a region and the number of buildings to present. We randomly take values for the co-ordinates of buildings. We then sort this according to left co-ordinate of the buildings. Sorting is done using a standard merge sort technique.

2. **Second step** We pass the array of buildings to the function.The base case is when there is only one building present, the function returns the co-ordinates of top left and bottom right points of the building. The function divides the array into two halves and recursively passes these two separate arrays to get the result. Combining these two skylines is done by going in a loop over the ascending order of x co-ordinates in either array. If the height of the current element is smaller than the last seen height of the other array, we update the height and add it to final array. Then another loop is run on the final array removing any redundant points, i.e. if the are on the same height as the previous points.

3. **Recurrence relation and time taken**
   The Recurrence relation is in the form of :
   $T(n) = 2 * T(n/2) + bn$ if $n > 1$
   $T(n) = a$ if $n = 1$
   Solving this relation, we can get the time taken for the function(taking into account, sorting) to be in the order of : $nlog(n)$

B.Skyline of a city with sloping rooftops

1. **First step** We take the input of a region and the number of buildings to present. We randomly take values for the co-ordinates of buildings. We then sort this according to left co-ordinate of the buildings. Sorting is done using a standard merge sort technique.

2. **Second step** We pass the array of buildings to the function.The base case is when there is only one building present, the function returns the co-ordinates of all 4 edges. The function divides the array into two halves and recursively

passes these two separate arrays to get the result. Combining these two skylines is done by going in a loop over the ascending order of x co-ordinates in either array. We start by adding the point to final array if it is higher than the curve, then finding an intersecting point(if it exists) and adding it to the array. Then another loop is run on the final array removing any redundant points, i.e. if the are on the same height as the previous points.

3. **Recurrence relation and time taken** he Recurrence relation is in the form of :
   $T(n) = 2 * T(n/2) + bn$ if $n > 1$
   $T(n) = a$ if $n = 1$
   Solving this relation, we can get the time taken for the function(taking into account, sorting) to be in the order of : $nlog(n)$