Problem 1

Solution.

If you draw \( n \) points in the plane, the maximal points form a staircase descending to the right. You can find them by different methods. One way is to scan the points from right to left. Another method is to use divide and conquer. That is, find the maximal points (staircase) on the left \( \frac{n}{2} \) points and the same for the right \( \frac{n}{2} \) points, and then merge the two staircases.

Problem 2

Solution.

One possible search algorithm in \( O(\log n) \) is binary search. But notice that in order to apply binary search, we need to bound the search area (in \( O(\log n) \)). We develop a method as follows:

Let \( i \) be an integer index of \( A \) and initially sets to be 1.
1. if \( A[2i] = \infty \), then look for \( x \) in the interval \( A[i \ldots 2i] \) using binary search.
2. else (\( A[2i] < \infty \)),
   a. if \( x < A[2i] \), then look for \( x \) in the interval \( A[i \ldots 2i] \) using binary search.
   b. else, set \( i \) to be \( 2i \), go to 1.

It is obvious that for the input of the binary search, the length is at most \( n \), so binary search takes \( O(\log n) \).

In order to locate the bound, for each step, we double the index \( i \). Therefore, after at most \( O(\log n) \) times, we can reach the end of sorted integers in \( A \). Above all, it takes \( O(\log n) \) time.