Activity Selection Problem

Following is the problem statement

You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

```
Example 1: Consider the following 3 activities sorted by by finish time.
     start[]  =  {10, 12, 20};
     finish[] =  {20, 25, 30};
A person can perform at most two activities. The maximum set of activities that can be executed is {0, 2}
[ These are indexes in start[] and finish[] ]

Example 2: Consider the following 6 activities sorted by by finish time.
     start[]  =  {1, 3, 0, 5, 8, 5};
     finish[] =  {2, 4, 6, 7, 9, 9};
A person can perform at most four activities. The maximum set of activities that can be executed is {0, 1, 3, 4}
[ These are indexes in start[] and finish[] ]
```

The greedy choice is to always pick the next activity whose finish time is least among the remaining activities and the start time is more than or equal to the finish time of previously selected activity. We can sort the activities according to their finishing time so that we always consider the next activity as minimum finishing time activity.

```
Algorithm:
    1) Sort the activities according to their finishing time
    2) Select the first activity from the sorted array and print it.
    3) Do following for remaining activities in the sorted array.
        3.1) If the start time of this activity is greater than or equal to the finish time of previously selecte
```

Time Complexity: It takes O(n log n) time if input activities may not be sorted. It takes O(n) time when it is given that input activities are always sorted.