Activity Selection problem is a approach of selecting non-conflicting tasks based on start and end time and can be solved in O(N logN) time using a simple greedy approach. Modifications of this problem are complex and interesting which we will explore as well. Suprising, if we use a Dynamic Programming approach, the time complexity will be O(N^3) that is lower performance.

```
Greedy-Iterative-Activity-Selector(A, s, f):



   Sort A by finish times stored in f

   S = {A[1]}

   k = 1



   n = A.length



   for i = 2 to n:

       if s[i] ≥ f[k]:

           S = S U {A[i]}

           k = i



   return S
```

Complexity

Time Complexity:

When activities are sorted by their finish time: `O(N)`

When activities are not sorted by their finish time, the time complexity is `O(N log N)` due to complexity of sorting