------------------- If you get the right strategy, your algorithm will complete the prove below -------------------

4) **PROOF of correctness for algorithm X** (that sorts the activities in the way answered in question 3)

Assume that the optimal set of activities is $a_1^*, a_2^*, a_3^*, \ldots, a_m^*$, and so the maximum number of activities possible is m. And this list of activities is already sorted in the increasing order of finish time.

Let the output of the algorithm X be $a_1^x, a_2^x, a_3^x, \ldots$. And also assume that this list of activities is sorted in the increasing order of finish time.

Then, assume further that $a_i^x = a_i^*$, for $i = 1,2,\ldots,j$. And then, $a_{j+1}^x <> a_{j+1}^*$

If a_{j+1}^x finishes later than a_{j+1}^*, would algorithm X picks a_{j+1}^x for its answer ?

It would have picked __a_{j+1}^*__ instead!

So, either a_{j+1}^x finishes earlier or at the same time as a_{j+1}^*.

Thus, we can replace a_{j+1}^* with a_{j+1}^x and the number of possible activities will remain to be __m__.

If we repeat the same consideration on the next activity of the two lists, a_{j+2}^* and a_{j+2}^x, we will find that we can replace a_{j+2}^* with a_{j+2}^x and the number of possible activities will remain to be __m__.

Is it possible that there is a $k < m$ such that a_{k+1}^x does not exist ?

Because a_k^x will finish __earlier__ than a_k^*, it will not conflict with a_{k+1}^*.

Therefore, at least algorithm X can at least pick __a_{k+1}^*__ for a_{k+1}^x, and thus a_{k+1}^x definitely exists.

By repeating the same consideration for the next value of k, it is clear that the number of activities produced by algorithm X must be at least __m__ .

But from assumption, the maximum number of activities possible is k, therefore, the number of activities produced by algorithm X must be at most __m__ .

Therefore, we can conclude that the number of activities produced by algorithm X is exactly __m__ .

And thus, the algorithm X gives the maximum number of activities possible … PROVED!

A GREEDY ALGORITHM IS MOSTLY VERY EFFICIENT, BUT **IT NEEDS TO BE PROVED FOR CORRECTNESS.**

Problem A : Activity Selection

A	1	2	3	4	5	6	7	8	9	10
start	19932	2591	9791	3849	2648	17034	29372	14826	15537	28277
finish	29111	18991	28235	13410	21288	19493	29791	25852	18867	29129

- sort Activity by finish time
- Go through all Activities
  if next start > previous finish :
  do count

29000

A	4	6	9	2	5	8	3	1	10	7	11
start	3849	17034	15537	2591	2648	14826	9791	19932	28277	29372	30000
finish	13410	19493	18867	18991	21288	25852	28235	29111	29129	29791	30001

a_1^* a_2^* a_3^* a_4^*

4 6 10 11

F=29129

$m=4$

$j=2$

$k=3$

F=29111

a_1^x a_2^x a_3^x a_4^x

4 6 1 7 Optimal?

∴ There is no way that a_3^x finish later than a_3^*

Same Thing for a_4^x a_4^*

Is it possible that a_{k+1}^x does not exist? $(k<m)$

$$a_k^x \text{ finish} \leq a_k^* \leq a_{k+1}^*$$

Then $a_{k+1}^* \longrightarrow a_{k+1}^x$