Classes, Classes!

DiPS CodeJam 22

Prompt

It's a long day, and Guru has a lot of activities to attend. He needs to select the maximum number of activities that he can do in a given time frame, assuming that he can only work on a single activity at a time. Each activity has a set start and end time.
Can you help him figure out how many activities he can attend?

Input Format

- The first line of the input contains an integer n, denoting the number of activities.
- The next n lines of the input each contain the start and end times of an activity, in the format $(\text{start}, \text{end})$.

Output Format

The first and only line of your output must contain a single integer m, denoting the maximum number of activities he can attend.

Constraints

- $4 \leq n \leq 24$
- Assume that the activities are already sorted based on end times.

Sample Input/Output

Input	Output
6 1 2 3 4 0 6 5 7 8 9 5 9	4

Solution

This is an example of the *Activity Selection Problem.*

Simplifying the Problem

Assume there exist n activities with each of them being represented by a start time s_i and finish time f_i. Two activities i and j are said to be non-conflicting if $s_i \geq f_j$ or $s_j \geq f_i$. The activity selection problem consists in finding the maximal solution set (S) of non-conflicting activities. Here, using a greedy algorithm to find the solution will always result in an optimal solution.

Solving the Problem

- Let us create an empty array **arr**.
- Now we can start adding activities to this array.
- Since this is a greedy algorithm, the first activity is always selected.
- Now we loop through the rest of the activities. For each activity:
  - If this activity has a start time that is greater than or equal to the finish time of the previously selected activity, then append it to **arr**.
- Finally, we print the length of **arr**, denoting the number of activities.

Sample Program

```
# n --> Total number of activities
# s[]--> An array that contains start time of all activities
# f[] --> An array that contains finish time of all activities

n = int(input())

s = []
f = []

for i in range(n):
     inputArr = list(map(int, input().split()))
     s.append(inputArr[0])
     f.append(inputArr[1])

activities = []

# The first activity is always selected
i = 0
activities.append(i)

# Consider rest of the activities
for j in range(n):

     # If this activity has start time greater than
     # or equal to the finish time of previously
     # selected activity, then select it
     if s[j] >= f[i]:
          activities.append(j)
          i = j

print(len(activities))
```