Q **Activity Selection Problem**

	a_1	a_2	a_3	a_4	a_5	a_6
start	1	3	0 ✗	5	8	5
end	2	4	6	7	9	9
	I	II		III	IV	

You are given n activities. Select the max no. of activities that can be performed by a single person.

ans → 4

Steps:

1. Sort the activities according to thier finishing time
2. Select the first activity from the sorted array.
3. For the remaining activities
   if (start [i] ≥ finish [prev]) use it

```
print Max activities (a, f, n) {
  → Sort on the basis of end time
  int i, j ;
  i=0;
  print (i);
  for (j=1 ; j<n; j++) {
     if (s[j] ≥ f[i] ) {
        print (j);
        i = j;
     }
  }
}
```

Q **Fractional knapsack**

Brute force approach: Try tout all subsets with all different fraction.

ans → [60, 100] [100, 20], [120, 30]
W → 50

Greedy approach:
1. Calculate the ratio of value/weight.
2. Sort the items based on density

(value / weight) ↑

60/100 , 100/20 , 120/30

6 5 4

60 + 100 + 80 → 240

Job Scheduling

Each job has a deadline and a profit associated.

Job	Deadline	Profit
a	4	20
b	1	10
c	1	40
d	1	30

1 job ≈ 1 day

→ Jobs can be performed in any order

→ Not necessary to perform all
→ Maximise profit

	d	p
a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

ans → 142
100 + 27 + 15

| c | a | e |

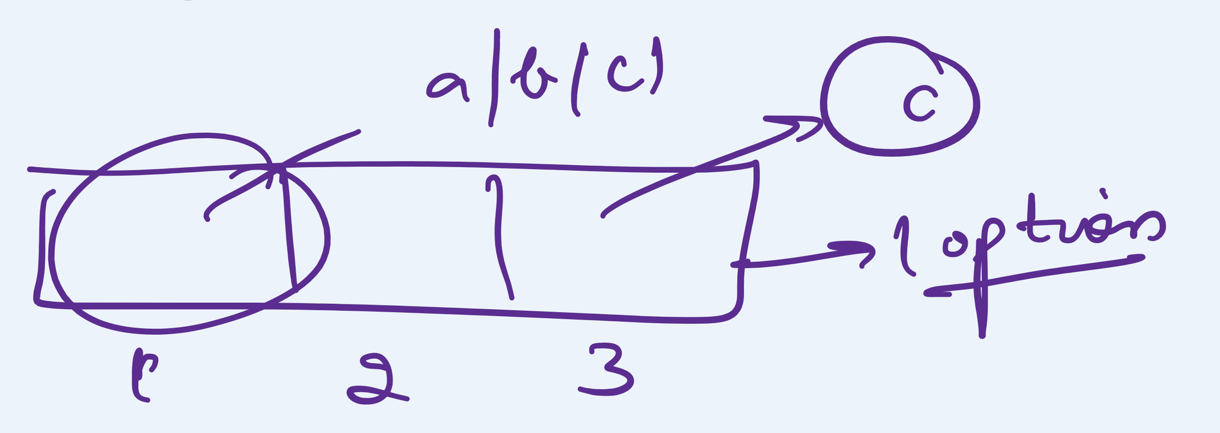
let's say

a	1	100
b	1	100
c	1	20

choose the job with max profit

now [a 2 100, b 1 100, c 3 100] xvx

what should be the order



For day 1, I have all options

Always try to perform a job on its deadline day.

1. which job to choose first ??
   → offers max profit.

2. when to prefer a job ??
   as close to deadline as possible

---

1. sort as per profit
2. Try to find a slot.

	d	p
→ a	2	100
b	1	19
→ c	2	27
d	1	25
→ e	3	15

c	a	e
1	2	3