problem 1 – Courses Allocation

Dynamic Programming Approach

(1) optimal substructure

The problem has optimal substructure.

If an optimal compatible set between activaties a_a and a_p contains an activity a_K, then:

The activites before a_K must be optimal between a_a and a_K

The activities after a_K must be optimal between a_K and a_p

Otherwise we could replace them with a better solution. contradicting optimality.

Therefore dynamic programming can be used.

(2) Top-down recursion (RecuSelect).

Idea:

Choose an activity that is compatible

Recursively solve left and right parts.

Pseudo description:

RecuSelect(s, f.k, n):

find first activity m that start after f[k]

if m exists:

return {a_m} ∪ RecuSelect(s, f, m, n)

else:

return empty set.

Start with:

RecuSelect(s, f, 0, n)

this builds a compatible schedule recursively.

(3) DP with tabulation

we create table c[a,p].

For each interval:

1. check all activites between them.

2. compute

$$c[a,p] = \max(c[a,k] + c[k,p] + 1)$$

Fill table from small intervals to large intervals.

Final answer: c[0,n]

Greedy Approach

1. Greedy choice

choose the activity with the earliest finishing time.

Reason:

This leaves maximum remaining time for future activities.

2. Greedy Pseudocode.

```
Greedy Schedule (s,f,n):
  sort activities by finish time.
  select a1
  last=1
  for i=2 to n:
   if s[i] >= f[last]:
     select ai
     last =i
  return selected activities
```

3. Proof of optimality

Induction

Base case:

First activity chosen has earliest finish, so it is safe.

inductive step:

Assume greedy is optimal for first k selections.

choosing next earliest finishing activity keeps maximum room, so remains optimal

stay ahead argument

At each step, greedy finishes no later than any other schedule.

Therefore it always leaves at least as much remaining time.

so greedy never falls behind optimal.

Contradiction

Assume there exists a better schedule

Replace its first activity with greedy's choice (earliest finish).

the schedule remains feasible and worse.

Repeat argument → contradiction.

thus greedy is optimal.