Job scheduling

profit = [15,27,10,100, 150]

jobs = ["j1", "j2", "j3", "j4", "j5"]

deadline = [2,3,3,3,4]

profitNJobs = list(zip(profit,jobs,deadline))

profitNJobs = sorted(profitNJobs, key = lambda x: x[0], reverse = True)

slot = []

for \_ in range(len(jobs)):

slot.append(0)

profit = 0

ans = []

for i in range(len(jobs)):

ans.append('null')

for i in range(len(jobs)):

job = profitNJobs[i]

#check if slot is occupied

for j in range(job[2], 0, -1):

if slot[j] == 0:

ans[j] = job[1]

profit += job[0]

slot[j] = 1

break

print("Jobs scheduled buddy:",ans[1:])

print(profit)

theory

Job Sequencing Problem – Theory

The Job Sequencing Problem is a classic optimization problem in computer science where the objective is to maximize total profit by scheduling a set of jobs within their given deadlines. Each job takes one unit of time and must be completed before or on its deadline.

🔹 Key Concepts:

Jobs have:

A unique identifier (e.g., j1, j2)

A profit if the job is completed on time

A deadline by which it must be completed

Constraints:

Each job takes exactly 1 time unit

A job must be completed on or before its deadline

Only one job can be executed at a time

Objective:

Select and schedule jobs in such a way that the total profit is maximized

🔹 Greedy Approach:

The problem is solved using a Greedy Algorithm:

Step 1: Sort all jobs in decreasing order of profit

Step 2: Initialize a list of available time slots

Step 3: For each job:

Try to schedule it in the latest available slot before or on its deadline

If the slot is available, assign the job and add its profit