**Name : Ayush Patil**

**C-3 Batch**

**Roll no : TYCOC210**

**Prn no : 122B1B210**

**Assignment -2**

**Title :**

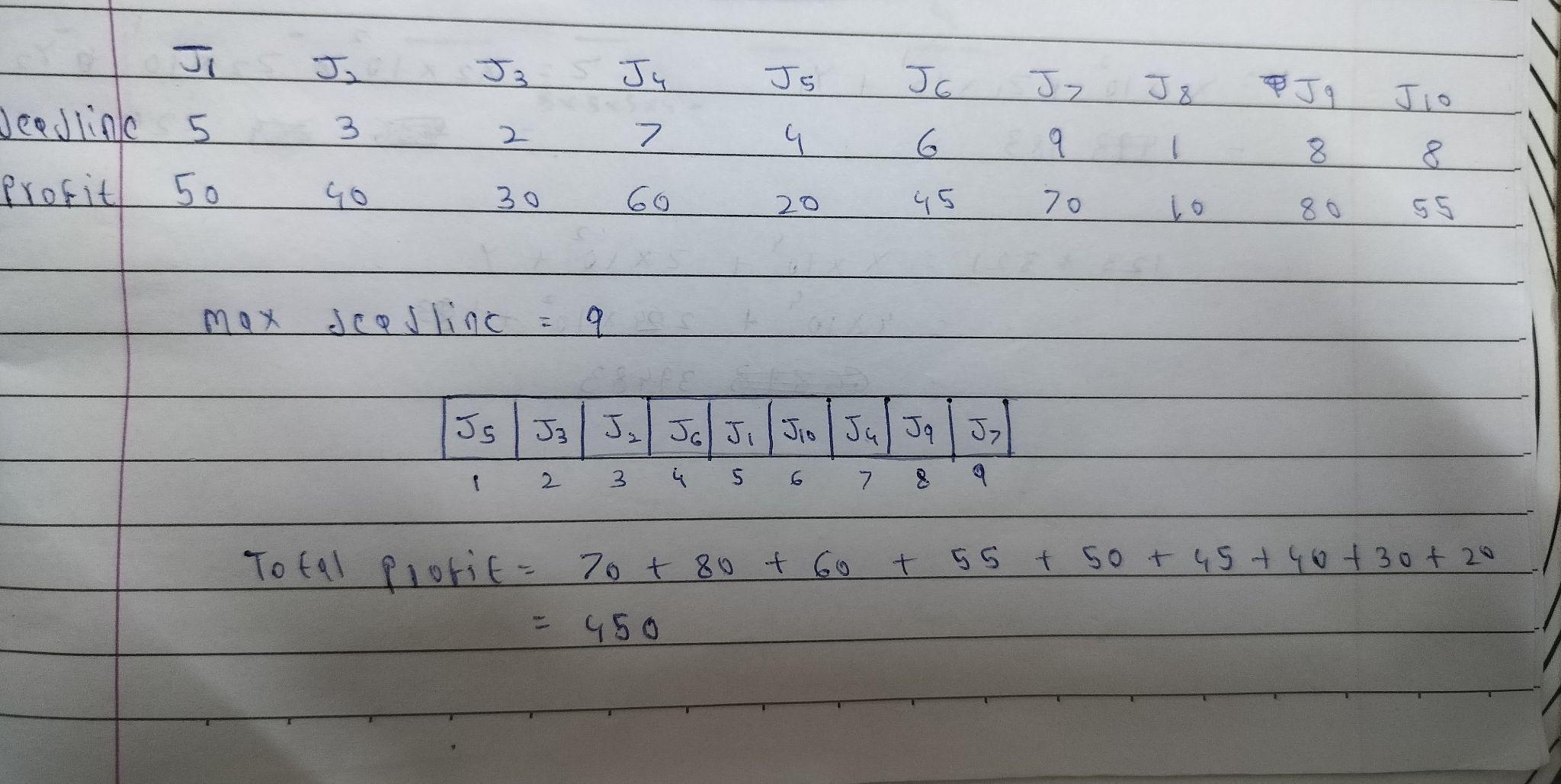
Consider the scheduling problem. n tasks to be scheduled on single processor. Let d1,..,dn be deadline and p1,….pn be the profit of each task to execute on single processor isknown. The tasks can be executed in any order but one task at a time and each task take 1unit of time to execute. Design a greedy algorithm

**Theory :**

Job scheduling algorithm is applied to schedule the jobs on a single processor to maximize the profits.The greedy approach of the job scheduling algorithm states that, “Given ‘n’ number of jobs with a starting time and ending time, they need to be scheduled in such a way that maximum profit is received within the maximum deadline”.

Time Complexity: O(N^2)

Auxiliary Space: O(N)



**CODE :**

class Task:

def \_\_init\_\_(self, profit, deadline):

self.profit = profit

self.deadline = deadline

def schedule\_tasks(tasks):

tasks.sort(key=lambda x: x.profit, reverse=True)

max\_deadline = max(task.deadline for task in tasks)

timeline = [None] \* max\_deadline

total\_profit = 0

for task in tasks:

for j in range(min(max\_deadline, task.deadline) - 1, -1, -1):

if timeline[j] is None:

timeline[j] = task

total\_profit += task.profit

break

return timeline, total\_profit

def main():

n = int(input("Enter the number of tasks: "))

tasks = []

for i in range(n):

profit = int(input(f"Enter profit for task {i + 1}: "))

deadline = int(input(f"Enter deadline for task {i + 1}: "))

tasks.append(Task(profit, deadline))

scheduled\_tasks, max\_profit = schedule\_tasks(tasks)

print("Total Profit:", max\_profit)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT :**

