

DAA Lab - Practical - 4

Name : Rishabh Jain

Roll No : A-54

Aim:

Implement activity selection and Possible Equal Maximum Sum Problem algorithm for the given scenario

Problem Statement:

PART (A)

In the single-machine scheduling problem, we are given a set of n activities A_i . Each job i has a starting time s_i , deadline d_i and profit p_i . At any time instant, we can do only one job. Doing a job i earns a profit p_i . Generate a solution to select the largest set of mutually compatible jobs and calculate the total profit generated by the machine. The greedy algorithm for single-machine scheduling selects the job using activity selection algorithm.

PART (B)

Given three stacks of the positive numbers, the task is to find the possible equal maximum sum of the stacks with the removal of top elements allowed.

▼ Part A -

```
start = [1,3,0,5,3,5,6,8,8,2,12]
end = [4,5,6,7,9,9,10,11,12,14,16]
act = ["a1","a2","a3","a4","a5","a6","a7","a8","a9","a10","a11"]
prof = [10,15,14,12,20,30,32,28,30,40,45]

final = [[start[0],end[0],act[0]]]
profit = prof[0]
k=1
for m in range(1,len(start)):
    if(start[m]>=end[k]):
        k=m
        profit = profit + prof[k]
        final.append([start[k],end[k],act[k]])

print("Profit - ",profit)
print("Final [start,end,activity name] - ",final)

Profit - 95
Final [start,end,activity name] - [[1, 4, 'a1'], [5, 7, 'a4'], [8, 11, 'a8'], [12, 16, 'a11']]

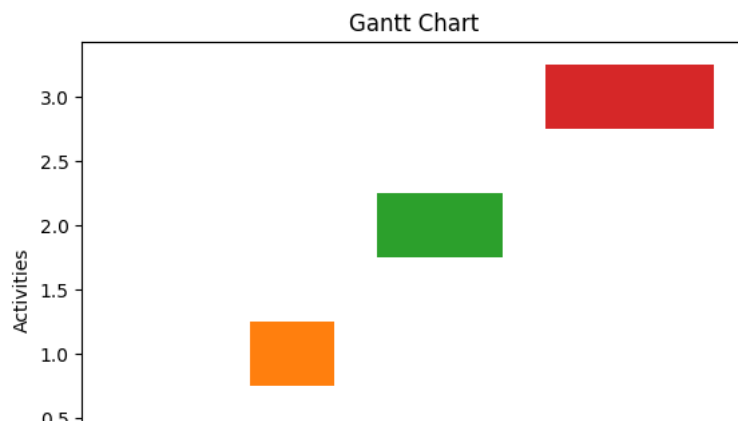
import matplotlib.pyplot as plt
import numpy as np

fig2, ax2 = plt.subplots()

for i in range(len(final)):
    start_Time = final[i][0]
    end_Time = final[i][1]
    ax2.barh(i, end_Time-start_Time, left=start_Time, height=0.5, align='center')

ax2.set_xlabel('Timeline')
ax2.set_ylabel('Activities')
ax2.set_title('Gantt Chart')

plt.show()
```



▼ Part B -

```
s1 = [3,5,8,5]
s2 = [2,2,4,9,6,5]
s3 = [2,1,2,3,5]
```

```
def pes():
    sum1 = sum(s1)
    sum2 = sum(s2)
    sum3 = sum(s3)

    while(1):
        if(sum1==sum2 and sum1==sum3):
            print(s1)
            print(s2)
            print(s3)
            break

        if(len(s1)==0 or len(s2)==0 or len(s3)==0):
            break

        if(sum1>sum2 and sum1>sum3):
            sum1 -= s1.pop()

        if(sum2>sum3 and sum2>sum1):
            sum2 -= s2.pop()

        if(sum3>sum2 and sum3>sum1):
            sum3 -= s3.pop()
```

```
pes()

[3, 5]
[2, 2, 4]
[2, 1, 2, 3]
```

