1) Compress the following string using Huffman Coding algorithm

a) DHEENADAYALAN

No. of character : 13

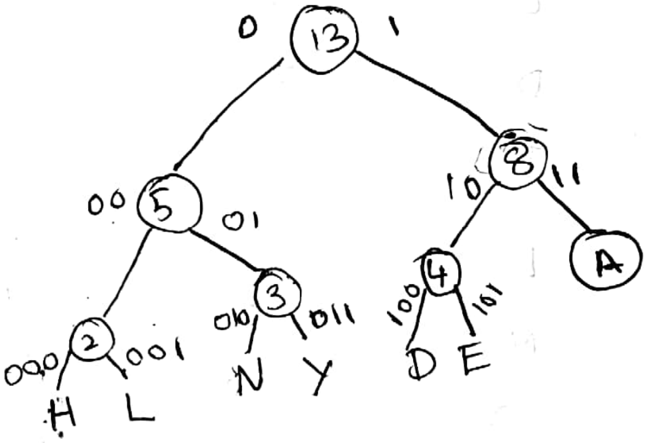
No. of bits required before compression : 13 × 2 = 104 bits

frequency of character

A B C D E F G H I J K L M N O P Q R S T

U V W X Y Z –

| Character | Frequency |
|---|---|
| A | 4 |
| D | 2 |
| E | 2 |
| H | 1 |
| L | 1 |
| N | 2 |
| Y | 1 |
| | 13 |



10 0 000 101101 010 11 100 11 011 11 001 11 010

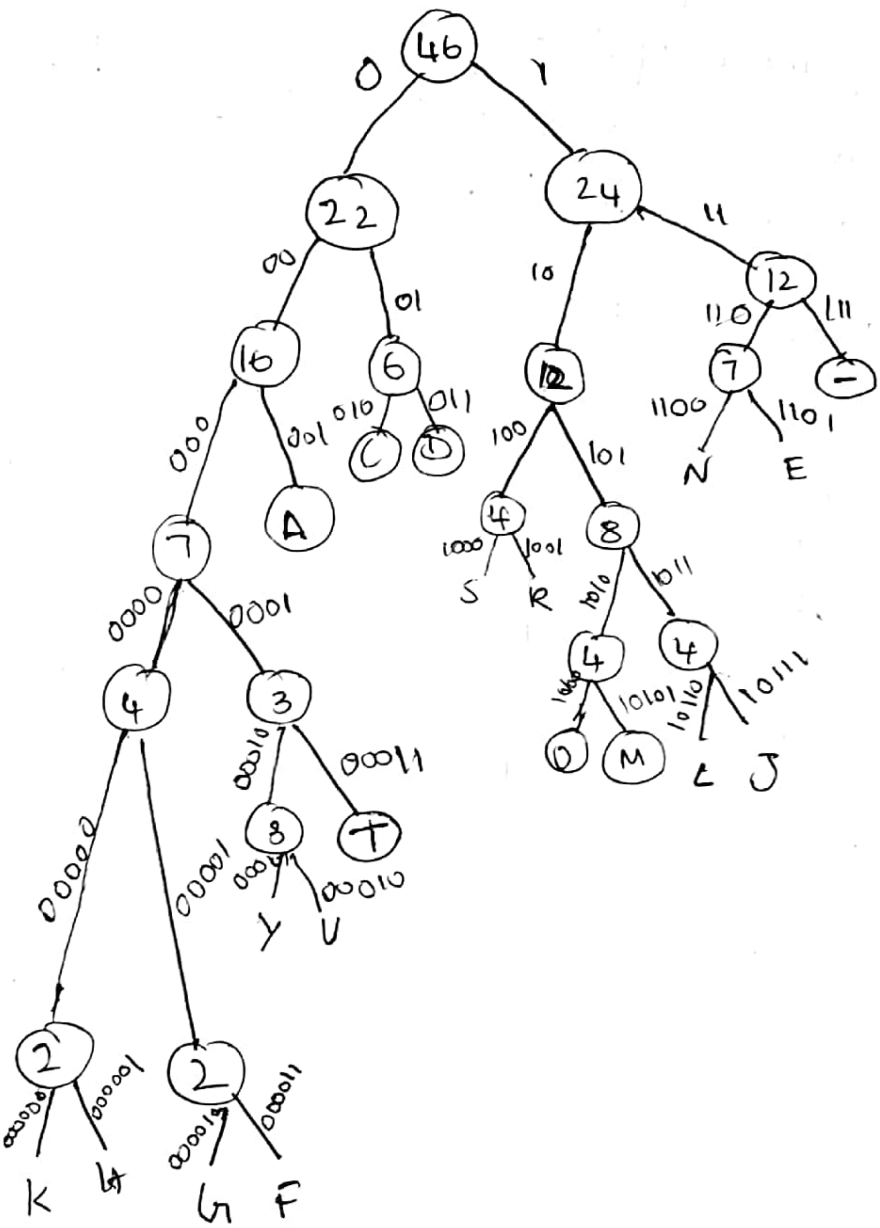
No. of bits required after compression 35 bits

5) KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

No. of Characters : 46

No. of bits required before Compression = 46×8 = 368 bits

| Character | frequency |
|---|---|
| A | 9 |
| – | 5 |
| E | 4 |
| N | 3 |
| C | 3 |
| D | 3 |
| I | 2 |
| L | 2 |
| M | 2 |
| O | 2 |
| R | 2 |
| S | 2 |
| T | 1 |
| U | 1 |
| Y | 1 |
| F | 1 |
| G | 1 |
| H | 1 |
| K | 1 |



00000000 11011000 11 0000011 011010 11111 0000001000

11 0101111 0010 1 000 10 1111 0 110101 000 100 1111 0101 000

1 00 1111 0101 000 100 1111 010 1000 100 1111 001 11011000 110

001 1001 0100000 1100111000 11111 1101011 000 101 1001

0001 0001+101111 01 01 1100 = 187

(3) complete ~~the~~ longest common Subse quence for the following string

a) RADIATION~~AND~~ and VARIATION

|   | E | V | A | R | I | A | T | I | O | N |
|---|---|---|---|---|---|---|---|---|---|---|
| E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0^ | 0^ | \1 | <1 | <1 | <1 | <1 | <1 | <1 |
| A | 0 | <0 | \1 | <1 | <2 | \2 | <2 | <2 | <2 | <2 |
| D | 0 | <0 | ^1 | ^1 | \2 | <2 | <2 | \3 | <3 | <3 |
| I | 0 | <0 | ^1 | ^1 | \2 | <2 | <2 | \3 | <3 | <3 |
| A | 0 | <0 | ^1 | <1 | ^2 | \3 | <3 | <3 | <3 | <3 |
| T | 0 | <0 | ^1 | ^1 | ^2 | ^3 | \4 | <4 | <4 | <4 |
| I | 0 | <0 | ^1 | ^1 | \2 | ^3 | ^4 | \5 | <5 | <5 |
| O | 0 | <0 | ^1 | <1 | ^2 | ^3 | ^4 | ^5 | \6 | <6 |
| N | 0 | <0 | ^1 | <1 | ^2 | ^3 | ^4 | ^5 | ^6 | \7 |

L.C.S of Radiation and Variation is 7'

5) ALGORITHMS and LOGARITHMS

| | E | L | O | G | A | R | I | T | H | M | S |
|---|---|---|---|---|---|---|---|---|---|---|---|
| E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 20 | <0 | <0 | \1 | <1 | <1 | <1 | <1 | <1 | <1 |
| L | 0 | \1 | <1 | <1 | \1 | <1 | <1 | <1 | <1 | <1 | <1 |
| G | 0 | ^1 | <1 | \2 | <2 | <2 | <2 | <2 | <2 | <2 | <2 |
| O | 0 | ^1 | \2 | <2 | <2 | <2 | <2 | <2 | <2 | <2 | <2 |
| R | 0 | ^1 | ^2 | <2 | <2 | \3 | <3 | <3 | <3 | <3 | <3 |
| I | 6 | ^1 | ^2 | <2 | <2 | ^3 | \4 | <4 | 44 | <4 | <4 |
| T | 0 | ^1 | ^2 | <2 | <2 | ^3 | ^4 | \5 | <5 | <5 | <5 |
| H | 0 | ^1 | ^2 | <2 | <2 | ^3 | 4^ | ^5 | \6 | <6 | <6 |
| M | 0 | ^1 | ^2 | <2 | <2 | ^3 | ^4 | ^5 | ^6 | \7 | <7 |
| S | 0 | ^1 | ^2 | ^2 | ^2 | ^3 | ^4 | ^5 | ^6 | ^7 | \8 |

logest . Common subre quences of logarithms and algorithms is 8

## ③ OBST

| Keys | Rank | Name | email | RegNO |
|---|---|---|---|---|
| frequency | 5 | 3 | 6 | 7 |

| j \ i | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 5 | $11^{(2)}$ | | |
| 1 | | 0 | 3 | $12^{(3)}$ | |
| 2 | | | 0 | 6 | |
| 3 | | | | 0 | 7 |
| 4 | | | | | 0 |

$$C(0,2) = \begin{cases} C(0,0) + C(1,2) + W(0,2) \\ C(0,1) + C(2,2) + W(0,2) \end{cases}$$

$$= \min \begin{cases} 0+3+8 = 4 \\ 5+0+5 = 10 \end{cases} = 11^{(1)}$$

$$C(1,3) = \min \begin{cases} C(1,1) + C(2,3) + W(1,3) \\ C(1,2) + C(3,3) + W(1,3) \end{cases}$$

$$= \min \begin{cases} 0+6+9 \\ 3+0+9 \end{cases} = 12^{(3)}$$

$$C(2,4) = \min \begin{cases} C(2,2) + C(3,4) + W(2,4) \\ C(2,3) + C(4,4) + W(2,4) \end{cases} = \min \begin{cases} 0+4+13 \\ 6+0+13 \end{cases}$$

$$= 19^{(6)}$$

$$C(0,3) = \min \begin{cases} C(0,0) + C(1,3) + W(0,3) \\ C(0,1) + C(2,3) + W(0,3) \\ C(0,2) + C(3,3) + W(0,3) \end{cases} = \min \begin{cases} 0+12+14 \\ 5+6+14 \\ 11+0+14 \end{cases}$$

$$= 25^{(3)}$$

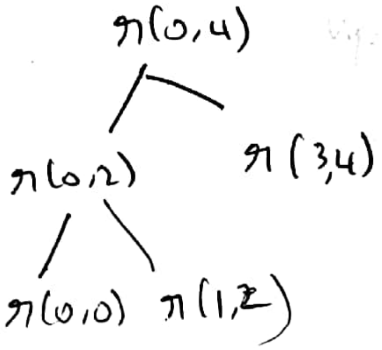
$$C(1,4) = \min \begin{cases} C(1,1) + C(2,4) + W(1,4) \\ C(1,2) + C(3,4) + W(1,8) \\ C(1,2) + C(4,4) + W(1,4) \end{cases} = \min \begin{cases} 0+19+16 \\ 3+7+16 \\ 25+0+18 \end{cases}$$

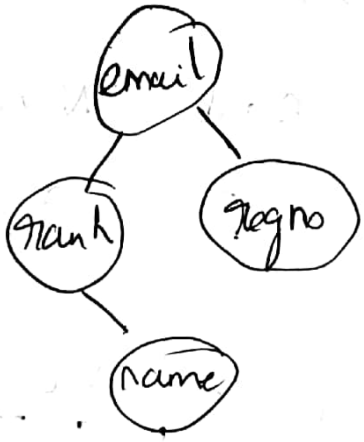
$$= 36^{(2)}$$

$$C(0,w) = \min \begin{cases} C(0,0) + C(1,4) + W(0,4) \\ C(0,1) + C(1,4) + W(0,4) \\ C(0,2) + C(3,4) + W(0,4) \\ C(0,3) + C(4,4) + W(0,4) \end{cases} = \min \begin{cases} 0+6+4 \\ \cancel{5+4+21} \\ 11+7+21 \\ 25+0+21 \end{cases}$$

$$= 39^{(3)}$$

obst





④

| Key | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| freq | 2 | 6 | 3 | 4 |

$$C(0,2) = \min \begin{cases} C(0,0) + C(1,2) + w(0,2) \\ C(0,1) + C(2,2) + w(0,2) \end{cases} = \min \begin{cases} 0+6+8 \\ 3+0+8 \end{cases} = 10^{(2)}$$

$$C(1,3) = \min \begin{cases} C(1,1) + C(2,3) + w(1,3) \\ C(2,3) + C(4,4) + w(2,4) \end{cases} = \min \begin{cases} 0+4+7 \\ 3+0+7 \end{cases}$$

$$C(0,3) = \min \begin{cases} C(0,0) + C(1,3) + w(0,3) \\ C(0,1) + C(2,3) + w(0,3) \\ C(0,2) + (3,3) + w(0,3) \end{cases} = \min \begin{cases} 0+12+11 \\ 2+3+11 \\ 10+0+11 \end{cases} = 16^{(1)}$$

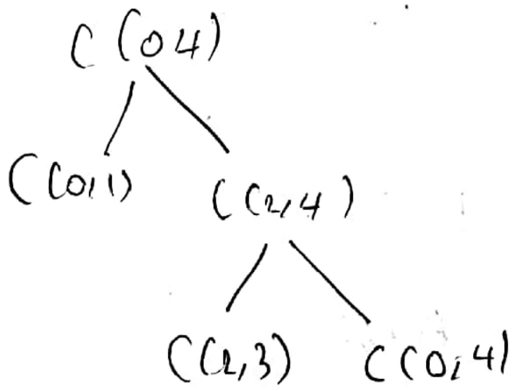
$$C(1,4) = \min \begin{cases} C(1,1) + C(2,4) + w(1,4) \\ C(1,2) + C(3,4) + w(1,4) \\ C(1,3) + C(4,4) + w(1,4) \end{cases} = \min \begin{cases} 0+10+13 \\ 10+4+13 \\ 16+0+13 \end{cases} = 23^{(2)}$$

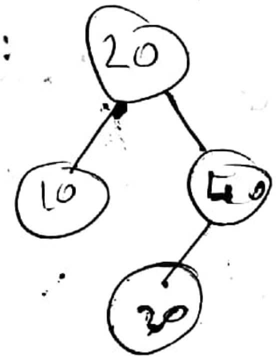
$$C(0,4) = \min \begin{cases} C(0,0) + C(1,4) + w(0,4) \\ C(0,1) + C(2,4) + w(0,4) \\ C(0,2) + C(3,4) + w(0,4) \\ C(0,3) + C(4,4) + w(0,4) \end{cases}$$

$$= \min \begin{cases} 0+23+15 \\ 2+10+15 \\ 10+4+15 \\ 16+0+15 \end{cases} = 27^{(2)}$$

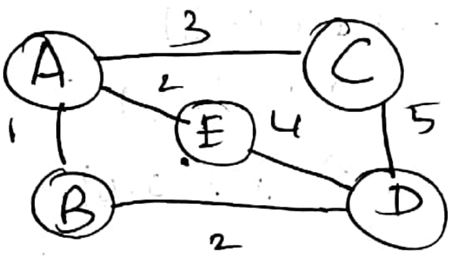
| j \ i | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 2 | $10^2$ | $16^2$ | $27^2$ |
| 1 |  | 0 | 6 | $12^2$ | $23^2$ |
| 2 |  |  | 0 | 3 | $10^4$ |
| 3 |  |  |  | 0 | 4 |
| 4 |  |  |  |  | 0 |

Optimal binary tree





## ⑤ Shortest path



| Source | A | B | C | D | E | Choose vertex |
|---|---|---|---|---|---|---|
| int{} | 0 | ∞ | ∞ | ∞ | ∞ | A |
| {A} | - | 1,A | 3,A | ∞ | 2,A | B |
| {A,B} | - | - | 3,A | 2,B<br>2,A | 2,A | E |
| {A,E} | - | 1,A | 3,A | 2,B<br>3,A | - | C |
| {A,C} | - | 1,A | - | 2,B<br>3,A | - | D |
| {A,B,C} | - | 1,A | 3,A | - | 2,A | - |

Result :

| Source | Destination | Path | Cost |
|---|---|---|---|
| A | B | AB | 1 |
| A | C | AC | 3 |
| A | D | ABD | 3 |
| A | E | AE | 2 |

# 6) Job Scheduling

| Tasks | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|
| Profit | 20 | 25 | 16 | 15 | 9 | 22 | 19 | 22 | 30 |
| deadline | 3 | 3 | 4 | 5 | 7 | 7 | 6 | 2 | 2 |

Soln:-

Desc order of Task

| Task | Profit | , deadline |
|---|---|---|
| T9 | 30 | 2 |
| T2 | 25 | 3 |
| T6 | 22 | 7 |
| T8 | 22 | 2 |
| T1 | 20 | 3 |
| T7 | 19 | 6 |
| T4 | 15 | 5 |
| T3 | 10 | 4 |
| T5 | 9 | 7 |

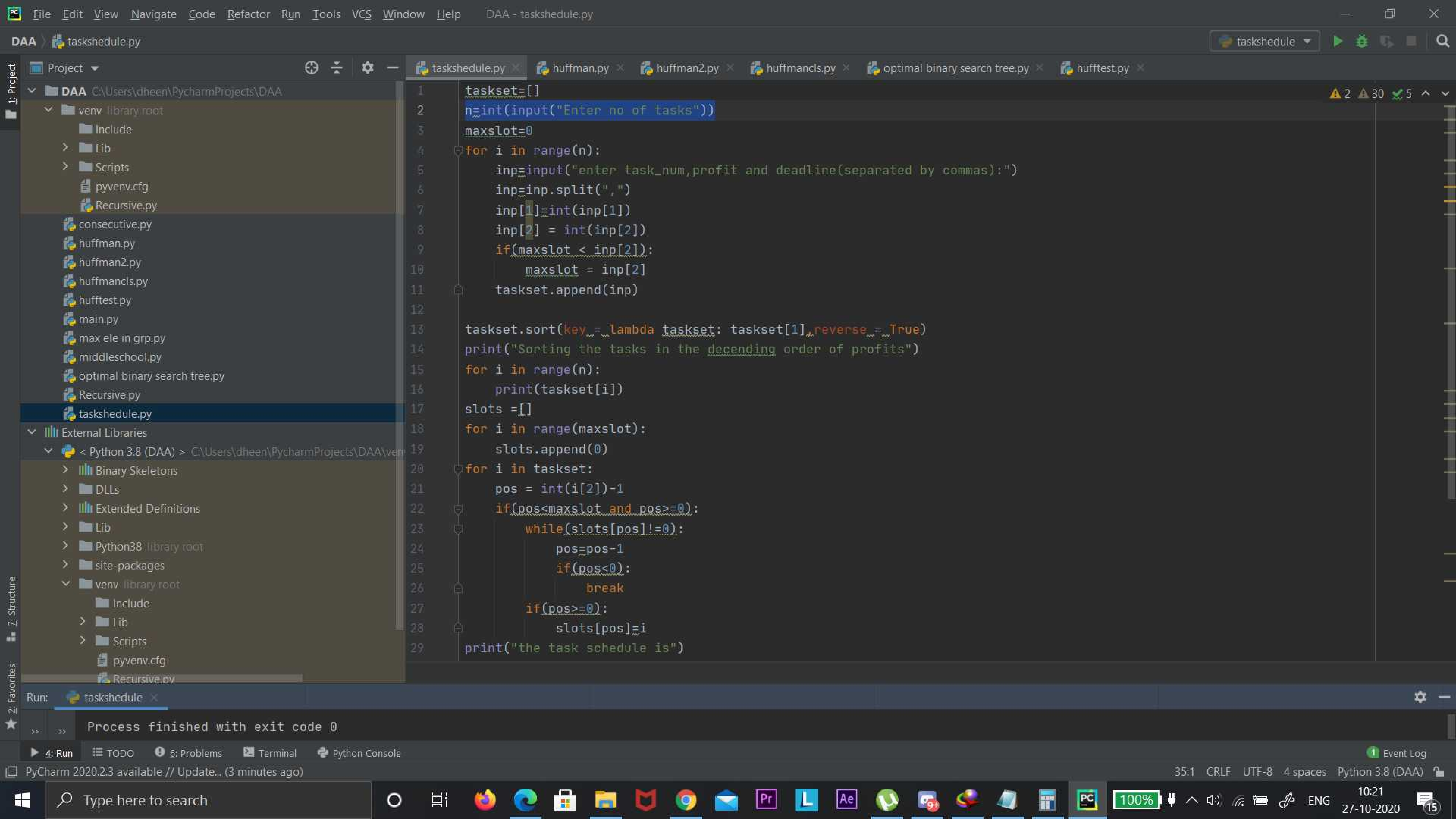
| t8 | t9 | t2 | t3 | t4 | t7 | t6 |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

optimal Schedule:

T8, T9, t2, t3, t4, t7 t6

Max Profit = 22+30+25+10+15 + 19+22

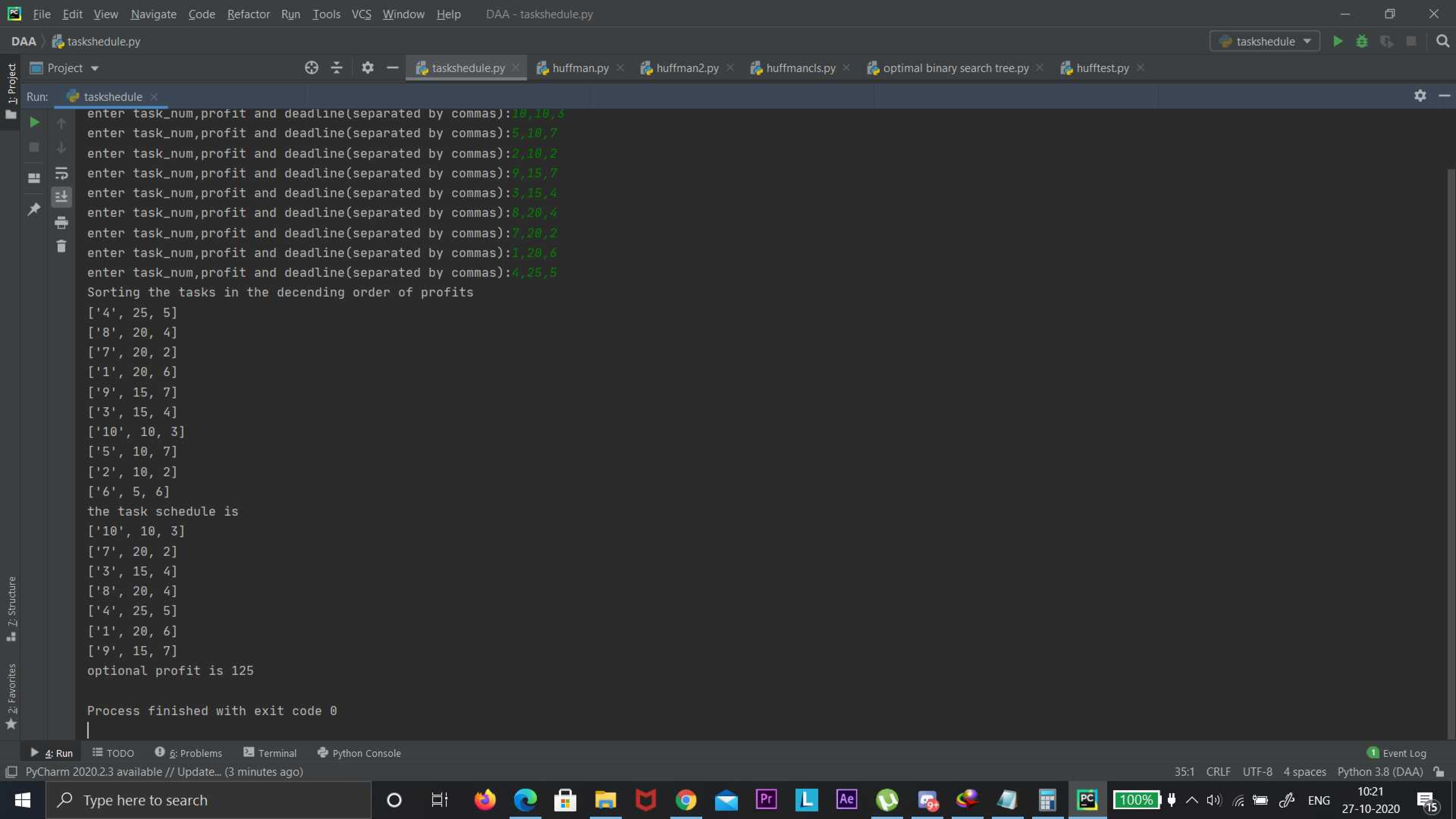
= 143



```python
taskset=[]
n=int(input("Enter no of tasks"))
maxslot=0
for i in range(n):
    inp=input("enter task_num,profit and deadline(separated by commas):")
    inp=inp.split(",")
    inp[1]=int(inp[1])
    inp[2] = int(inp[2])
    if(maxslot < inp[2]):
        maxslot = inp[2]
    taskset.append(inp)

taskset.sort(key = lambda taskset: taskset[1],reverse = True)
print("Sorting the tasks in the decending order of profits")
for i in range(n):
    print(taskset[i])
slots =[]
for i in range(maxslot):
    slots.append(0)
for i in taskset:
    pos = int(i[2])-1
    if(pos<maxslot and pos>=0):
        while(slots[pos]!=0):
            pos=pos-1
            if(pos<0):
                break
        if(pos>=0):
            slots[pos]=i
print("the task schedule is")
```

```
Process finished with exit code 0
```



PyCharm 2020.2.3 available // Update... (3 minutes ago)

```
enter task_num,profit and deadline(separated by commas):10,10,3
enter task_num,profit and deadline(separated by commas):5,10,7
enter task_num,profit and deadline(separated by commas):2,10,2
enter task_num,profit and deadline(separated by commas):9,15,7
enter task_num,profit and deadline(separated by commas):3,15,4
enter task_num,profit and deadline(separated by commas):8,20,4
enter task_num,profit and deadline(separated by commas):7,20,2
enter task_num,profit and deadline(separated by commas):1,20,6
enter task_num,profit and deadline(separated by commas):4,25,5
Sorting the tasks in the decending order of profits
['4', 25, 5]
['8', 20, 4]
['7', 20, 2]
['1', 20, 6]
['9', 15, 7]
['3', 15, 4]
['10', 10, 3]
['5', 10, 7]
['2', 10, 2]
['6', 5, 6]
the task schedule is
['10', 10, 3]
['7', 20, 2]
['3', 15, 4]
['8', 20, 4]
['4', 25, 5]
['1', 20, 6]
['9', 15, 7]
optional profit is 125

Process finished with exit code 0
```