Project 3

Karthik Raj U2320917J
Ishita Jain U2321668B
Jayakumar Thirunithiyan U2321424H

Question 1

Give a recursive definition of the function P(C).

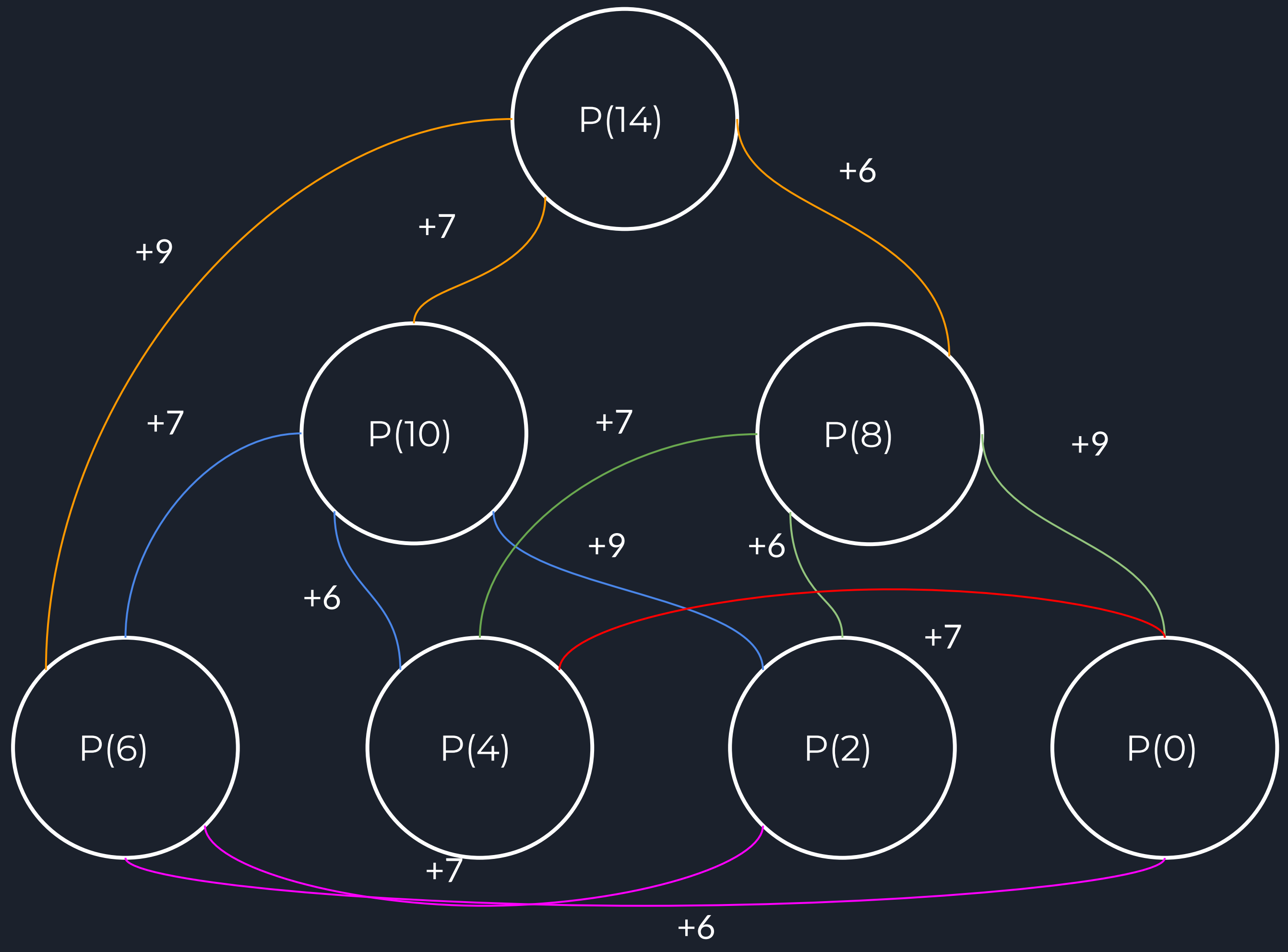
$P(C) = 0$, where C or $n = 0$

$P(C) = \max(P(C), p(C\text{-weight}[i]) + P(i));$, $\text{weight}[i] < C$

Question 2

Draw the subproblem graph for P(14) where n is 3 with the weights and profits given below.

	0	1	2
w_i	4	6	8
p_i	7	6	9


P(14)
+9
+7
+6
P(10)
P(8)
+7
+7
+9
+9
+6
+6
+7
P(6)
P(4)
P(2)
P(0)
+7
+6

Question 3

Give a dynamic programming algorithm to compute the maximum profit, given a knapsack of capacity C, n types of objects with weights wi and profits pi using the bottom up approach.

```
def knapsack(C, weight, profit, n):
    # Initialise the DP table
    # Values are all initialised to 0 but in actuality, only row 0 and col 0 will be 0
    V = [[0 for i in range(C + 1)] for i in range(n + 1)]

    # Construct the table in a bottom up manner
    for i in range(1, n + 1): # Iterate through the ROWS (Dictates the capacity limit we set)
        for w in range(1, C + 1): # Iterate through the COLS (Dictates the number of objects we can consider)
            if weight[i-1] <= w: # If the weight of object is less than the capacity limit
                # Max value of the two possible options
                # 1: We DO NOT include the i'th object -> V[i-1][w]
                # 2: Even if we include that instance, still can include its multiple instance
                V[i][w] = max(V[i-1][w], V[i][w-weight[i-1]] + profit[i-1])
            else:
                V[i][w] = V[i-1][w]
    return V[n][C], V
```

Question 4

Code your algorithm in a programming language

a) show the running result of P(14) with weights and profits given in (2).

	0	1	2
w_i	4	6	8
p_i	7	6	9

a. show the running result of P(14) with weights and profits given in (2).

wi = [4,6,8] pi = [7,6,9]

```
wi = [4,6,8]
pi = [7,6,9]

data = {
  "Weight(i)": [4, 6, 8],
  "Profit(i)": [7, 6, 9]
}

df = pd.DataFrame(data).T
df.head()
```

	0	1	2
Weight(i)	4	6	8
Profit(i)	7	6	9

```
capacity = 14
max_profit,table1 = knapsack(capacity,wi,pi,len(wi))

t1 = pd.DataFrame(table1)
print("The Maximum Profit is: "+ str(max_profit))
t1.head()
```

	0	1	2
Weight(i)	4	6	8
Profit(i)	7	6	9

wi = [4,6,8]
pi = [7,6,9]
n = 3
P = 14

The Maximum Profit is: 21

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	7	7	7	7	14	14	14	14	21	21	21
2	0	0	0	0	7	7	7	7	14	14	14	14	21	21	21
3	0	0	0	0	7	7	7	7	14	14	14	14	21	21	21

Question 4

Code your algorithm in a programming language
b) Show the running result of P(14) with weights and profits given below.

	0	1	2
w_i	5	6	8
p_i	7	6	9

b. Show the running result of P(14) with weights and profits given below.

wi = [5,6,8] pi = [7,6,9]

```
w2i = [5,6,8]

data2 = {
  "Weight(i)": [5, 6, 8],
  "Profit(i)": [7, 6, 9]
}

df2 = pd.DataFrame(data2).T
df2.head()
```

	0	1	2
Weight(i)	5	6	8
Profit(i)	7	6	9

```
max_profit2,table2 = knapsack(capacity,w2i,pi,len(w2i))

t2 = pd.DataFrame(table2)
print("The Maximum Profit is: "+ str(max_profit2))
t2.head()
```

	0	1	2
Weight(i)	5	6	8
Profit(i)	7	6	9

wi = [5,6,8]
pi = [7,6,9]
n = 3
P = 14

```
The Maximum Profit is: 16
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	7	7	7	7	7	14	14	14	14	14
2	0	0	0	0	0	7	7	7	7	7	14	14	14	14	14
3	0	0	0	0	0	7	7	7	9	9	14	14	14	16	16