CZ2101 Project 3
BCS3 Group 5

Presented by:
Lin Jiajun (U2021181L)
Deionna Chee Rui Ping (U2021449D)
Fabrianne Effendi (U2021488E)

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

Assuming $C >= \max(w_i)$,

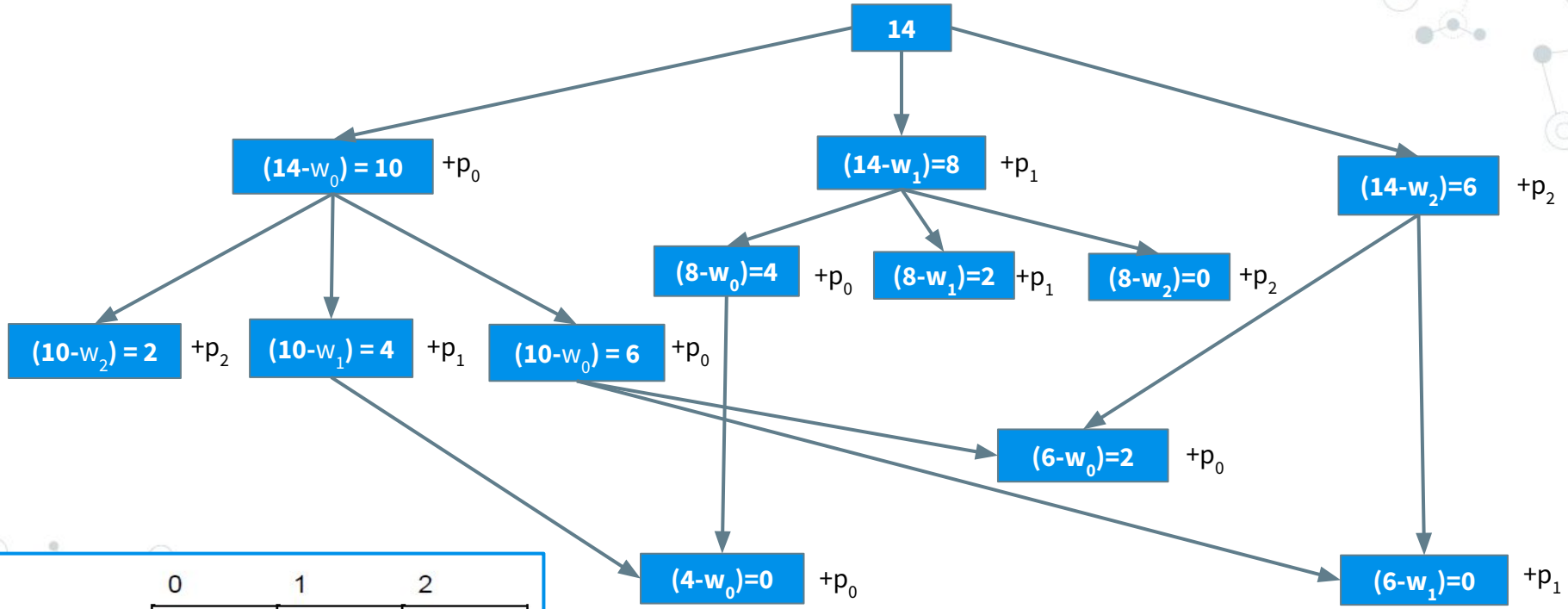
$P(C) = \max(P(C), p_0 + P(C-w_0), p_1 + P(C-w_1), \ldots, p_{n-1} + P(C-w_{n-1}))$

Final recursive definition:

$P(C) = \max(P(C), P(C - w[j]) + p[j])$

Where j varies from 0 to n-1 such that $w[j] <= C$

(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

(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, n, pi, wi):
    for (i = 0 to C):
        P[i] = 0
    for (i = 0 to C):
        for (j = 0 to (n-1)):
            if (wi[j] <= i):
                P[i] = max(P[i], P[i - wi[j]] + pi[j])
    return P[C]
```

(4) Code your algorithm in a programming language

```
def knapSack(C, n, price, weight):
    P = [0 for i in range(C + 1)]

    for i in range(C + 1):
        for j in range(n):
            if (weight[j] <= i):
                P[i] = max(P[i], P[i - weight[j]] + price[j])
    print("Price at each capacity C is given by:")
    print(P)
    return P[C]


C = int(input("Enter capacity of knapsack: "))
n = int(input("Enter number of items: "))
price = list(map(int, input("Value of items: ").split()))
weight = list(map(int, input("Weight of items: ").split()))

print("Maximum profit:",knapSack(C,n,price,weight))
```

P stores maximum value with knapsack capacity i

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

Function inputs

- price = [7,6,9]
- weight = [4,6,8]

Input

```
Enter capacity of knapsack: 14
Enter number of items: 3
Value of items: 7 6 9
Weight of items: 4 6 8
```

Output

Max profit = 7 + 7 + 7 = **21**

Output

```
Price at each capacity C is given by:
[0, 0, 0, 0, 7, 7, 7, 7, 14, 14, 14, 14, 21, 21, 21]
Maximum profit: 21
```

	0	1	2
w_i	4	6	8
p_i	7	6	9

```
Price at each capacity C is given by:
[0, 0, 0, 0, 7, 7, 7, 7, 14, 14, 14, 14, 21, 21, 21]
Maximum profit: 21
```

Capacity = 0-3: **Max Profit = 0**

Capacity = 4-7: **Max Profit = 7**

Capacity = 8-11: **Max Profit = 7+7 = 14**

Capacity = 12-14: **Max Profit = 7+7+7 = 21**

	0	1	2
w_i	4	6	8
p_i	7	6	9

Output: **Max Profit = 21**

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

Function inputs

- price = [7,6,9]
- weight = [5,6,8]

Input:

```
Enter capacity of knapsack: 14
Enter number of items: 3
Value of items: 7 6 9
Weight of items: 5 6 8
```

Output

Max profit = 7 + 9 = **16**

Output:

```
Price at each capacity C is given by:
[0, 0, 0, 0, 0, 7, 7, 7, 9, 9, 14, 14, 14, 16, 16]
Maximum profit: 16
```

	0	1	2
w_i	5	6	8
p_i	7	6	9

```
Price at each capacity C is given by:
[0, 0, 0, 0, 0, 7, 7, 7, 9, 9, 14, 14, 14, 16, 16]
Maximum profit: 16
```

Capacity = 0-4: **Max Profit = 0**

Capacity = 5-7: **Max Profit = 7**

Capacity = 8-9: **Max Profit = 9**

Capacity = 10-12: **Max Profit = 7 + 7 = 14**

Capacity = 13-14: **Max Profit = 7 + 9 = 16**

	0	1	2
w_i	5	6	8
p_i	7	6	9

Output: **Max Profit = 16**

	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	7	7	7
5	0	7	7	7
6	0	7	7	7
7	0	7	7	7

	0	1	2
w_i	4	6	8
p_i	7	6	9

```
Price at each capacity C is given by:
[0, 0, 0, 0, 7, 7, 7, 7, 14, 14, 14, 14, 21, 21, 21]
Maximum profit: 21
```

	0	1	2	3
8	0	7	14	14
9	0	7	14	14
10	0	7	14	14
11	0	7	14	14
12	0	7	14	21
13	0	7	14	21
14	0	7	14	**21**

	0	1	2
w_i	4	6	8
p_i	7	6	9

Output: Max Profit = 21

```
Price at each capacity C is given by:
[0, 0, 0, 0, 7, 7, 7, 7, 14, 14, 14, 14, 21, 21, 21]
Maximum profit: 21
```

	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	7	7	7
6	0	7	7	7
7	0	7	7	7

	0	1	2
w_i	5	6	8
p_i	7	6	9

```
Price at each capacity C is given by:
[0, 0, 0, 0, 0, 7, 7, 7, 9, 9, 14, 14, 14, 16, 16]
Maximum profit: 16
```

	0	1	2	3
8	0	7	9	9
9	0	7	9	9
10	0	7	14	14
11	0	7	14	14
12	0	7	14	14
13	0	7	14	16
14	0	7	14	**16**

	0	1	2
w_i	5	6	8
p_i	7	6	9

Output: Max Profit = 16

```
Price at each capacity C is given by:
[0, 0, 0, 0, 0, 7, 7, 7, 9, 9, 14, 14, 14, 16, 16]
Maximum profit: 16
```