CS-463 HW5

Bruno Francisco

Revision: May 02, 2021

1. Let $F(j)$ be the solution for a rod of length j and sale prices $p_1, p_2, \ldots, p_j$. The recurrence relation is:

   $$F(j) = \max\{F(j-1) + p_1, F(j-2) + p_2, \ldots, F(j-j) + p_j\}$$

   The answer to the original problem is returned when $j = n$. That is:

   $$F(n) = \max\{F(n-1) + p_1, F(n-2) + p_2, \ldots, F(n-n) + p_n\}$$

   The base case is given by a rod with length $j = 0$ whose solution is:

   $$F(0) = 0$$

   Pseudocode for the algorithm:

```
F[0] = 0
for j=1 to n do:
    max = p_1 + F[j-1]
    for i=2 to j do:
        curr = p_i + F[j-i]
        if curr > max do:
            max = curr
    F[j] = max
return F[n]
```

   The runtime is:

   $$\sum_{j=1}^{n}\sum_{i=2}^{j} 1$$

   $$= \sum_{j=1}^{n}(j-1)$$

   $$= \frac{n(n+1)}{2} - n$$

   Thus, the time complexity is $\Theta(n^2)$.

2. Let $F(i,j)$ be the solution for a $i \times j$ board. The recurrence relation is:

$$F(i,j) = F(i-1,j) + F(i,j-1)$$

The answer to the original $n \times m$ board is:

$$F(n,m) = F(n-1,m) + F(n,m-1)$$

When the board is only a single column or a single row there can only be one path, leading to the following base cases:

$$F(i,1) = 1 \text{ for all } 1 \leqslant i \leqslant n$$
$$F(1,j) = 1 \text{ for all } 1 \leqslant j \leqslant m$$

Pseudocode for the algorithm:

```
for i=1 to n do:
    F[i, 1] = 1
for j=1 to m do:
    F[1, j] = 1
for i=2 to n do:
    for j=2 to m do:
        F[i, j] = F[i-1, j] + F[i, j-1]
return F[n, m]
```

The runtime is:

$$\sum_{i=1}^{n} 1 + \sum_{j=1}^{m} 1 + \sum_{i=2}^{n} \sum_{j=2}^{m} 1$$

$$= n + m + \sum_{i=2}^{n} (m-1)$$

$$= n + m + (n-1)(m-1)$$

Therefore, the time complexity is $\Theta(nm)$.

Algorithm applied on a 5×6 board:

	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	2	3	4	5	6
3	1	3	6	10	15	21
4	1	4	10	20	35	56
5	1	5	15	35	70	126

Hence, there are 126 possible paths to the bottom-right position on a 5×6 board.

3. Let $F(i, j)$ be the the maximum money obtainable from the bottom edge to the square (i, j). The recurrence relation is:

$F(i, j) = \max\{F(i-1, j) + p((i-1, j), (i, j)), F(i-1, j+1) + p((i-1, j+1), (i, j)), F(i-1, j-1) + p((i-1, j-1), (i, j))\}$

The answer to the original problem will be the maximum value of all n squares in the top edge. That is:

$\max\{F(n, 1), F(n, 2), \ldots, F(n, n)\}$

The base cases occur when we reach the bottom square. That is, the base cases are:

$F(1, j) = 0$ for all $1 \leqslant j \leqslant n$

Pseudocode for the algorithm:

```
for j=1 to n do:
    F[1, j] = 0
for i=2 to n do:
    for j=1 to n do:
        case1 = F[i-1, j] + p([i-1, j], [i, j])
        case2 = case1
        if j > 1 do:
            case2 = F[i-1, j-1] + p([i-1, j-1], [i, j])
        case3 = case1
        if j < n do:
            case3 = F[i-1, j+1] + p([i-1, j+1], [i, j])
        maxCase = case1
        if case2 > max do:
            maxCase = case2
        if case3 > max do:
            maxCase = case3
        F[i, j] = maxCase
maxValue = F[n, 1]
for j=2 to n do:
    if F[n, j] > maxValue do:
        maxValue = F[n, j]
return maxValue
```

The runtime is:

$$\sum_{j=1}^{n} 1 + \sum_{i=2}^{n} \sum_{j=1}^{n} 1 + \sum_{j=2}^{n} 1$$

$$= n + \sum_{i=2}^{n} n + (n-1)$$

$$= n(n-1) + 2n - 1$$

$$= n^2 - n + 2n - 1$$
$$= n^2 + n - 1$$

Therefore, the time complexity is $\Theta(n^2)$.

4. Calculation:

$$F(1,1) = F(1-1,1) = F(0,1) = 0$$
$$F(1,2) = \max\{F(1-1,2), F(1-1,2-2)+20\} = \max\{0,20\} = 20$$
$$F(1,3) = \max\{F(1-1,3), F(1-1,3-2)+20\} = \max\{0,20\} = 20$$
$$F(1,4) = \max\{F(1-1,4), F(1-1,4-2)+20\} = \max\{0,20\} = 20$$
$$F(1,5) = \max\{F(1-1,5), F(1-1,5-2)+20\} = \max\{0,20\} = 20$$
$$F(1,6) = \max\{F(1-1,6), F(1-1,6-2)+20\} = \max\{0,20\} = 20$$
$$F(1,7) = \max\{F(1-1,7), F(1-1,7-2)+20\} = \max\{0,20\} = 20$$
$$F(2,1) = F(2-1,1) = 0$$
$$F(2,2) = F(2-1,2) = 20$$
$$F(2,3) = \max\{F(2-1,3), F(2-1,3-3)+25\} = \max\{20,25\} = 25$$
$$F(2,4) = \max\{F(2-1,4), F(2-1,4-3)+25\} = \max\{20,25\} = 25$$
$$F(2,5) = \max\{F(2-1,5), F(2-1,5-3)+25\} = \max\{20,45\} = 45$$
$$F(2,6) = \max\{F(2-1,6), F(2-1,6-3)+25\} = \max\{20,45\} = 45$$
$$F(2,7) = \max\{F(2-1,7), F(2-1,7-3)+25\} = \max\{20,45\} = 45$$
$$F(3,1) = \max\{F(3-1,1), F(3-1,1-1)+15\} = \max\{0,15\} = 15$$
$$F(3,2) = \max\{F(3-1,2), F(3-1,2-1)+15\} = \max\{20,15\} = 20$$
$$F(3,3) = \max\{F(3-1,3), F(3-1,3-1)+15\} = \max\{25,35\} = 35$$
$$F(3,4) = \max\{F(3-1,4), F(3-1,4-1)+15\} = \max\{25,40\} = 40$$
$$F(3,5) = \max\{F(3-1,5), F(3-1,5-1)+15\} = \max\{45,40\} = 40$$
$$F(3,6) = \max\{F(3-1,6), F(3-1,6-1)+15\} = \max\{45,60\} = 60$$
$$F(3,7) = \max\{F(3-1,7), F(3-1,7-1)+15\} = \max\{45,60\} = 60$$
$$F(4,1) = F(4-1,1) = 15$$
$$F(4,2) = F(4-1,2) = 20$$
$$F(4,3) = F(4-1,3) = 35$$
$$F(4,4) = \max\{F(4-1,4), F(4-1,4-4)+40\} = \max\{40,40\} = 40$$
$$F(4,5) = \max\{F(4-1,5), F(4-1,5-4)+40\} = \max\{40,55\} = 55$$
$$F(4,6) = \max\{F(4-1,6), F(4-1,6-4)+40\} = \max\{60,60\} = 60$$
$$F(4,7) = \max\{F(4-1,7), F(4-1,7-4)+40\} = \max\{60,75\} = 75$$

$$F(5,1) = F(5-1,1) = 15$$
$$F(5,2) = F(5-1,2) = 20$$
$$F(5,3) = F(5-1,3) = 35$$
$$F(5,4) = F(5-1,4) = 40$$
$$F(5,5) = \max\{F(5-1,5), F(5-1,5-5)+35\} = \max\{55,35\} = 55$$
$$F(5,6) = \max\{F(5-1,6), F(5-1,6-5)+35\} = \max\{60,50\} = 60$$
$$F(n,W) = F(5,7) = \max\{F(5-1,7), F(5-1,7-5)+35\} = \max\{75,55\} = 75$$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	20	20	20	20	20	20
2	0	0	20	25	25	45	45	45
3	0	15	20	35	40	40	60	60
4	0	15	20	35	40	55	60	75
5	0	15	20	35	40	55	60	75

The subset includes items 1, 3, and 4, with a total value of 75 dollars.