**Rod Cutting**

Given a rod of length n and prices of piece of rod length i, where 1 <= i <= n . Find the optimal way to cut the given rod into smaller rods in order to maximize profit.

e.g.

Given a rod of length 4m and the prices of different lengths of rods

| Length (l) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Price (pi) | 1 | 5 | 8 | 9 | 10 | 17 | 17 | 20 | 24 | 30 |

Possibilities

- We don't cut the given rod of length 4 (9)
- We cut the rod into 4 pieces (3 cuts), each of length 1 (1 + 1 + 1 + 1 = 4)
- We cut the rod into 2 pieces (1 cut), one of length 1 and other of length 3 (1 + 8 = 9)
- We cut the rod into 3 pieces (2 cuts), two pieces of length 1 and other of length 2 (1 + 1 + 5 = 7)
- We cut the rod into 2 pieces (1 cut), each of length 2 (5 + 5 = 10)

e.g.

Length of the Rod is 5m. **n = 5m , Prices = {2, 5, 7, 3}**

To get to the solution to the actual problem (given rod of length 5m and prices of rod length 1m(2$), 2m(5$), 3m(7$), 4m(3$), what is the maximum profit we can get?

We will first solve the subproblems.

e.g.

• Given rod of length 1m and when we can only make cut of 1m, what is the maximum profit we can get (answer is 2$ {1m} )

• Given rod of length 4m and when we can only make cut of 1m, what is the maximum profit we can get (answer is 8$ {1m, 1m, 1m, 1m} )

• Given rod of length 2m and when we can only make cut of 1m and 2m, what is the maximum profit we can get (answer is 5$ {2m} )

• Given rod of length 4m and when we can make cut of 1m, 2m, 3m and 4m, what is the maximum profit we can get (answer is 10$ {2m, 2m} )

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | | | | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

When we make a cut of 0 meter, the maximum profit we can make is 0$.

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Similarly when the rod length is 0 meter regardless of the different cuts we are allowed to make, the maximum profit we can make is 0$.

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| 1 | **0** | | | | | |
| 2 | **0** | | | | | |
| 3 | **0** | | | | | |
| 4 | **0** | | | | | |

**When we can only make cuts of 1m**

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| 1 | **0** | **2** | **4** | **6** | **8** | **10** |
| 2 | **0** | | | | | |
| 3 | **0** | | | | | |
| 4 | **0** | | | | | |

**When we can only make cuts of 1m and 2m**

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 4 | 6 | 8 | 10 |
| 2 | 0 | 2 | 5 | 7 | 10 | 12 |
| 3 | 0 | | | | | |
| 4 | 0 | | | | | |

**When we can only make cuts of 1m, 2m and 3m**

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 4 | 6 | 8 | 10 |
| 2 | 0 | 2 | 5 | 7 | 10 | 12 |
| 3 | 0 | 2 | 5 | 7 | 10 | 12 |
| 4 | 0 | | | | | |

**When we can make cuts of 1m, 2m, 3m and 4m**

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 4 | 6 | 8 | 10 |
| 2 | 0 | 2 | 5 | 7 | 10 | 12 |
| 3 | 0 | 2 | 5 | 7 | 10 | 12 |
| 4 | 0 | 2 | 5 | 7 | 10 | 12 |

When we can't make $i^{th}$ meter cut from the rod of length j meter , because i > j

dpTable[ i ][ j ] = dpTable[ i - 1] [ j ]

When we can make $i^{th}$ meter cut from the rod of length j meter , because i is not greater than j ( i <= j )

dpTable[ i ][ j ] = Max( dpTable[ i - 1] [ j ] , prices[ i - 1] + dpTable[ i ] [ j – i ] )

**n = 5m , Prices = {2, 5, 7, 3}**

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| 1 | **0** | **2** | **4** | **6** | **8** | **10** |
| 2 | **0** | **2** | **5** | **7** | **10** | **12** |
| 3 | **0** | **2** | **5** | **7** | **10** | **12** |
| 4 | **0** | **2** | **5** | **7** | **10** | **12** |

dpTable[2][1] = Since rod length is only 1m and we can't make cut of 2m from a rod of length just 1 m, dpTable[2][1] = dpTable[1][1] = 2

dpTable[2][2] = Max ( dpTable[1][2] , 5 + dpTable[2][2-2] ) = Max ( 4 , 5 ) = 5

dpTable[2][3] = Max ( dpTable[1][3] , 5 + dpTable[2][3-2] ) = Max ( 6 , 7 ) = 7

dpTable[2][4] = Max ( dpTable[1][4] , 5 + dpTable[2][4-2] ) = Max ( 8 , 10 ) = 10

dpTable[2][5] = Max ( dpTable[1][5] , 5 + dpTable[2][5-2] ) = Max ( 10 , 12 ) = 12

dpTable[3][3] = Max ( dpTable[2][3] , 7 + dpTable[3][3-3] ) = Max ( 7 , 7 ) = 7

dpTable[3][4] = Max ( dpTable[2][4] , 7 + dpTable[3][4-3] ) = Max ( 10 , 9 ) = 10

dpTable[4][5] = Max ( dpTable[3][5] , 3 + dpTable[4][5-4] ) = Max ( 12 , 5 ) = 12

**But what cuts we made?**

n = 5m , Prices = {2, 5, 7, 3}

| Cut/Length | 0(m) | 1(m) | 2(m) | 3(m) | 4(m) | 5(m) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 4 | 6 | 8 | 10 |
| 2 | 0 | 2 | 5 | 7 | 10 | 12 |
| 3 | 0 | 2 | 5 | 7 | 10 | 12 |
| 4 | 0 | 2 | 5 | 7 | 10 | 12 |

**Cuts : 2m, 2m, 1m**

**Implementation**

```
public class App {
	public static void main(String[] args) {
		int[] prices = { 2, 5, 7, 3 };
		int rodLength = 5;
		RodCutting rodCutting = new RodCutting(prices, rodLength);
		int maximumProfit = rodCutting.solve();
		System.out.println("Maximum Profit : " + maximumProfit);
		rodCutting.showCuts();
	}
}
```

```java
public class RodCutting {
	int [] prices;
	int rodLength;
	int[][] dpTable;
	public RodCutting(int[] prices, int rodLength) {
		this.prices = prices;
		this.rodLength = rodLength;
		dpTable = new int[prices.length + 1][rodLength + 1];
	}

	public int solve() {
		....
	}

	public void showCuts() {
		....
	}
}
```

```java
public int solve() {
    for (int i = 1; i <= prices.length; i++) {
       for (int j = 1; j <= rodLength; j++) {
		if (i > j) {
			dpTable[i][j] = dpTable[i - 1][j];
		} else {
			dpTable[i][j] = Math.max(dpTable[i - 1][j],
                                                prices[i - 1] + dpTable[i][j - i]);
		}
	 }
     }
    return dpTable[prices.length][rodLength];
}
```

```java
public void showCuts() {
   for(int i= prices.length, j=rodLength;  i > 0; ) {
	if(dpTable[i][j] != 0 && dpTable[i][j] != dpTable[i-1][j]) {
		System.out.println("Make a cut of " + i+"m");
		j = j -i;
	 }else {
		i--;
	}
    }
}
```