

## تمرین ۱. سوال تئوری

فرض کنید شما یک آرایه شامل مقادیر مختلفی سکه دارید و قصد دارید عدد  $x$  را با جمع تعدادی از این سکه ها بسازید. (دقت کنید که از هر نوع سکه تعداد بینهایت در دسترس است)

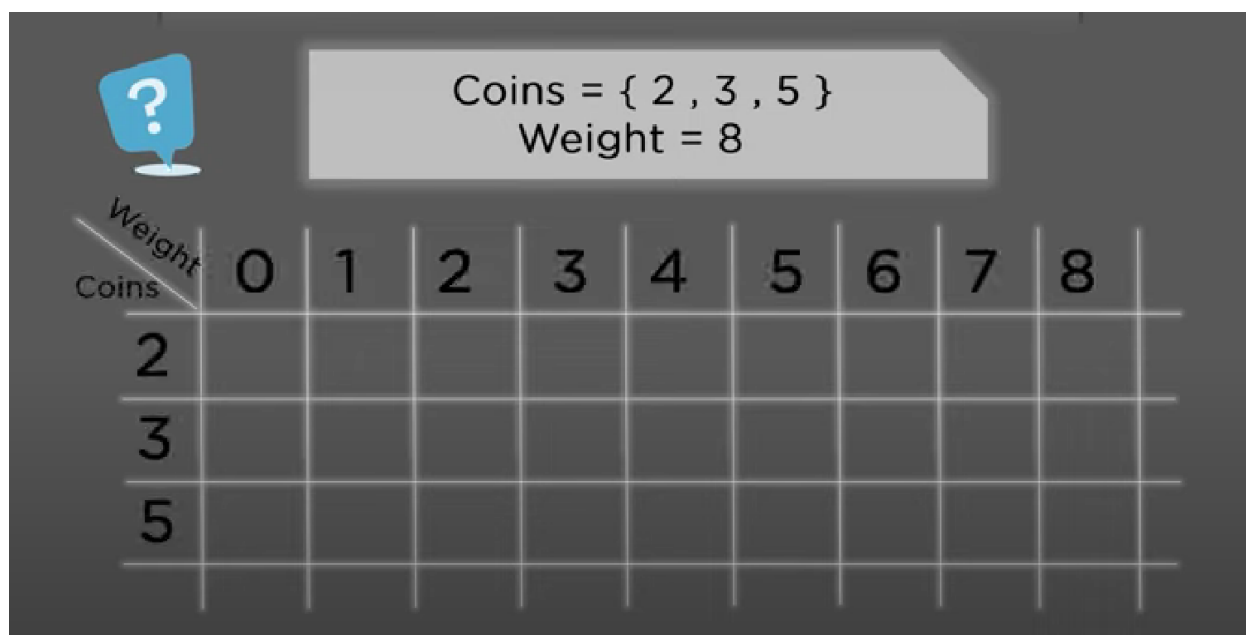
الگوریتمی مبتنی بر برنامه نویسی پویا ارائه دهید تا تعداد روشهای منحصر بفرد انتخاب این سکه ها که مجموع آنها برابر عدد  $X$  شود را خروجی دهد.

برای مثال اگر آرایه سکه های ما به شکل  $\{1, 2, 5\}$  باشد و مقدار  $x=5$  باشد تعداد حالت های مورد نظر ما برابر 4 میشود.

$\{1,1,1,1,1\}, \{5\}, \{2,2,1\}, \{2,1,1,1\}$

الف) الگوریتم خود را به طور کامل شرح دهید و پیچیدگی زمانی و حافظه آن را مشخص کنید.

آرایه ی 2 بعدی ای درست میکنیم که تعداد ستون هایش  $x+1$  (0 تا  $x$ ) و تعداد ردیف هایش هم تعداد سکه هایی که داریم میگذاریم.



|                |   |                       |   |   |   |   |   |   |   |   |
|----------------|---|-----------------------|---|---|---|---|---|---|---|---|
|                |   | Coins = { 2 , 3 , 5 } |   |   |   |   |   |   |   |   |
|                |   | Weight = 8            |   |   |   |   |   |   |   |   |
| Coins \ Weight |   | 0                     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|                | 2 |                       |   |   |   |   |   |   |   |   |
|                | 3 |                       |   |   |   |   |   |   |   |   |
|                | 5 |                       |   |   |   |   |   |   |   |   |

ستون اول که مقدار 0 باید با سکه ها پر شود را با 1 پر میکنیم چون یک حالت بیشتر ندارد.

| Weight \ Coins | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----------------|---|---|---|---|---|---|---|---|---|
| 2              | 1 |   |   |   |   |   |   |   |   |
| 3              | 1 |   |   |   |   |   |   |   |   |
| 5              | 1 |   |   |   |   |   |   |   |   |

برای هر خانه (مثلا  $coin[x]$ ) باید حساب کنیم مجموع حالاتی که آن سکه باشد (باقی مانده  $1 - coin[x]$  که میشود خانه ای در همان ردیف اما با ستونی متفاوت) + آن سکه نباشد (که میشود خانه ی بالایی اش). این عمل را از بالای جدول شروع میکنیم و ردیف به ردیف پر میکنیم. خانه یی که آخرین  $weight$  و  $coin$  است جواب مسئله ی ما است. (به زبان دیگر میشود جمع تعداد حالاتی که آن سکه هست با حالاتی که نیست ، که تمام حالات تشکیل میدهد). مثال:

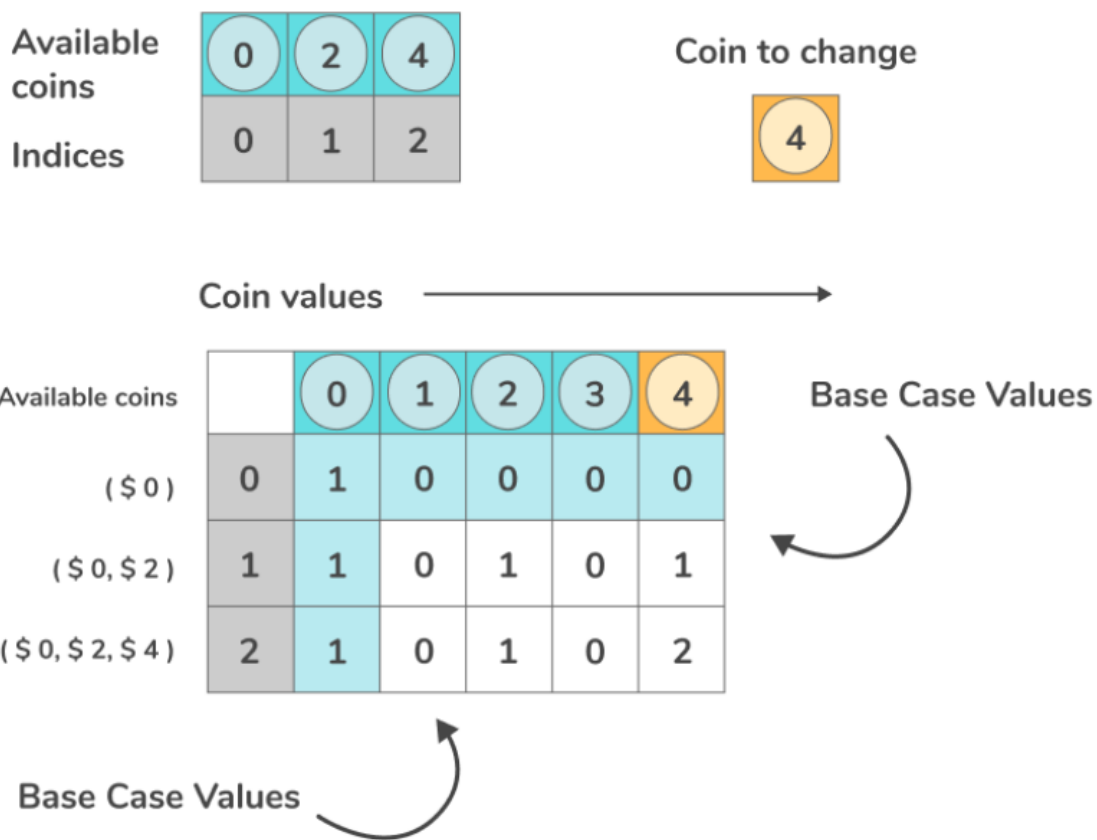
| Weight \ Coins | 0 | 1 | 2 | 3       | 4       | 5       | 6 | 7 | 8 |
|----------------|---|---|---|---------|---------|---------|---|---|---|
| 2              | 1 | 0 | 1 | 0       | 1       | 0       | 1 | 0 | 1 |
| 3              | 1 | 0 | 1 | $0+1=1$ | $1+0=1$ | 1       | 2 | 1 | 2 |
| 5              | 1 | 0 | 1 | 1       | 1       | $1+1=2$ |   |   |   |

Exclude coin 5 ( copy the above value of coin 2 and 3 =  $0 + 1 = 1$ ) + include coin 5 ( weight of 5 - coin value 5 = 0 and weight of 0 at coin value 5)

| Weight<br>Coins | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------------|---|---|---|---|---|---|---|---|---|
| 2               | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 3               | 1 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 2 |
| 5               | 1 | 0 | 1 | 1 | 1 | 2 | 3 | 2 | 3 |

پ.ن: بهتر است ردیف دیگری هم بالای سکه ی 2 داشته باشیم و آنرا با 0 پر کنیم، اینطوری برای ردیف سکه ی 2 در دسر کمتری داریم در کد زدنش.

مثال دیگر:



coin\_change[i][j] // coin change is the name of matrix i is denotes the value of coin and j denotes the weight

```
for( i =0 ; i <= coins.length ; i++ )
{
    for ( j =0 ; j <=weight ; j++ )
    {
        if ( coins[i] > j )
            coin_change[i][j] = coin_change[i-1][j]
        else
            coin_change[i][j]= coin_change[i-1][j] + coin_change[i][j-
coins[i]
    }
}
```

exclude Only

exclude + include

Time complexity:  $O(\text{numberOfCoins} * \text{TotalAmount})$

Space complexity:  $O(\text{numberOfCoins} * \text{TotalAmount})$

ب) الگوریتم را بر روی مقادیر سکه  $x = \{30, 25, 5, 10\}$  پیاده سازی کنید. (جدول dp مورد نظر را کشیده و پر کنید. شیوه پیمایش آن را شرح دهید و جواب نهایی را ارائه دهید).

با فرض اینکه سکه هابه ترتیب  $\{25, 5, 10\}$  و مقدار هدف ما 30 است، جدول اینگونه پر میشود.

| 00                 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |    |
|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 01                 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 25 |    |
| 01                 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 02 | 5  |
| 01                 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 03 | 00 | 00 | 00 | 00 | 04 | 00 | 00 | 00 | 00 | 05 | 10 |
| Total Solutions: 5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Total Solutions: 5

PS: D:\code-ex\DP\DP2

نحوه پر شدن جدول:

```

Filled dynamicprogTable[1][1] = 0
Filled dynamicprogTable[1][2] = 0
Filled dynamicprogTable[1][3] = 0
Filled dynamicprogTable[1][4] = 0
Filled dynamicprogTable[1][5] = 0
Filled dynamicprogTable[1][6] = 0
Filled dynamicprogTable[1][7] = 0
Filled dynamicprogTable[1][8] = 0
Filled dynamicprogTable[1][9] = 0
Filled dynamicprogTable[1][10] = 0
Filled dynamicprogTable[1][11] = 0
Filled dynamicprogTable[1][12] = 0
Filled dynamicprogTable[1][13] = 0
Filled dynamicprogTable[1][14] = 0
Filled dynamicprogTable[1][15] = 0
Filled dynamicprogTable[1][16] = 0
Filled dynamicprogTable[1][17] = 0
Filled dynamicprogTable[1][18] = 0
Filled dynamicprogTable[1][19] = 0
Filled dynamicprogTable[1][20] = 0
Filled dynamicprogTable[1][21] = 0
Filled dynamicprogTable[1][22] = 0
Filled dynamicprogTable[1][23] = 0
Filled dynamicprogTable[1][24] = 0
Filled dynamicprogTable[1][25] = 0 + 1
Filled dynamicprogTable[1][26] = 0 + 0
Filled dynamicprogTable[1][27] = 0 + 0
Filled dynamicprogTable[1][28] = 0 + 0
Filled dynamicprogTable[1][29] = 0 + 0
Filled dynamicprogTable[1][30] = 0 + 0
Filled dynamicprogTable[2][1] = 0
Filled dynamicprogTable[2][2] = 0
Filled dynamicprogTable[2][3] = 0
Filled dynamicprogTable[2][4] = 0
Filled dynamicprogTable[2][5] = 0 + 1
Filled dynamicprogTable[2][6] = 0 + 0
Filled dynamicprogTable[2][7] = 0 + 0
Filled dynamicprogTable[2][8] = 0 + 0
Filled dynamicprogTable[2][9] = 0 + 0
Filled dynamicprogTable[2][10] = 0 + 1
Filled dynamicprogTable[2][11] = 0 + 0
Filled dynamicprogTable[2][12] = 0 + 0
Filled dynamicprogTable[2][13] = 0 + 0
Filled dynamicprogTable[2][14] = 0 + 0
Filled dynamicprogTable[2][15] = 0 + 1
Filled dynamicprogTable[2][16] = 0 + 0
Filled dynamicprogTable[2][17] = 0 + 0
Filled dynamicprogTable[2][18] = 0 + 0
Filled dynamicprogTable[2][19] = 0 + 0
Filled dynamicprogTable[2][20] = 0 + 1
Filled dynamicprogTable[2][21] = 0 + 0
Filled dynamicprogTable[2][22] = 0 + 0
Filled dynamicprogTable[2][23] = 0 + 0
Filled dynamicprogTable[2][24] = 0 + 0
Filled dynamicprogTable[2][25] = 1 + 1
Filled dynamicprogTable[2][26] = 0 + 0
Filled dynamicprogTable[2][27] = 0 + 0
Filled dynamicprogTable[2][28] = 0 + 0
Filled dynamicprogTable[2][29] = 0 + 0
Filled dynamicprogTable[2][30] = 0 + 2
Filled dynamicprogTable[3][1] = 0
Filled dynamicprogTable[3][2] = 0
Filled dynamicprogTable[3][3] = 0
Filled dynamicprogTable[3][4] = 0
Filled dynamicprogTable[3][5] = 1
Filled dynamicprogTable[3][6] = 0
Filled dynamicprogTable[3][7] = 0
Filled dynamicprogTable[3][8] = 0
Filled dynamicprogTable[3][9] = 0
Filled dynamicprogTable[3][10] = 1 + 1
Filled dynamicprogTable[3][11] = 0 + 0
Filled dynamicprogTable[3][12] = 0 + 0
Filled dynamicprogTable[3][13] = 0 + 0
Filled dynamicprogTable[3][14] = 0 + 0
Filled dynamicprogTable[3][15] = 1 + 1
Filled dynamicprogTable[3][16] = 0 + 0
Filled dynamicprogTable[3][17] = 0 + 0
Filled dynamicprogTable[3][18] = 0 + 0
Filled dynamicprogTable[3][19] = 0 + 0
Filled dynamicprogTable[3][20] = 1 + 2
Filled dynamicprogTable[3][21] = 0 + 0
Filled dynamicprogTable[3][22] = 0 + 0
Filled dynamicprogTable[3][23] = 0 + 0
Filled dynamicprogTable[3][24] = 0 + 0
Filled dynamicprogTable[3][25] = 2 + 2
Filled dynamicprogTable[3][26] = 0 + 0
Filled dynamicprogTable[3][27] = 0 + 0
Filled dynamicprogTable[3][28] = 0 + 0
Filled dynamicprogTable[3][29] = 0 + 0
Filled dynamicprogTable[3][30] = 2 + 3

```

[ویدیو توضیح الگوریتم](#)

[کد الگوریتم و توضیحات بیشتر](#)