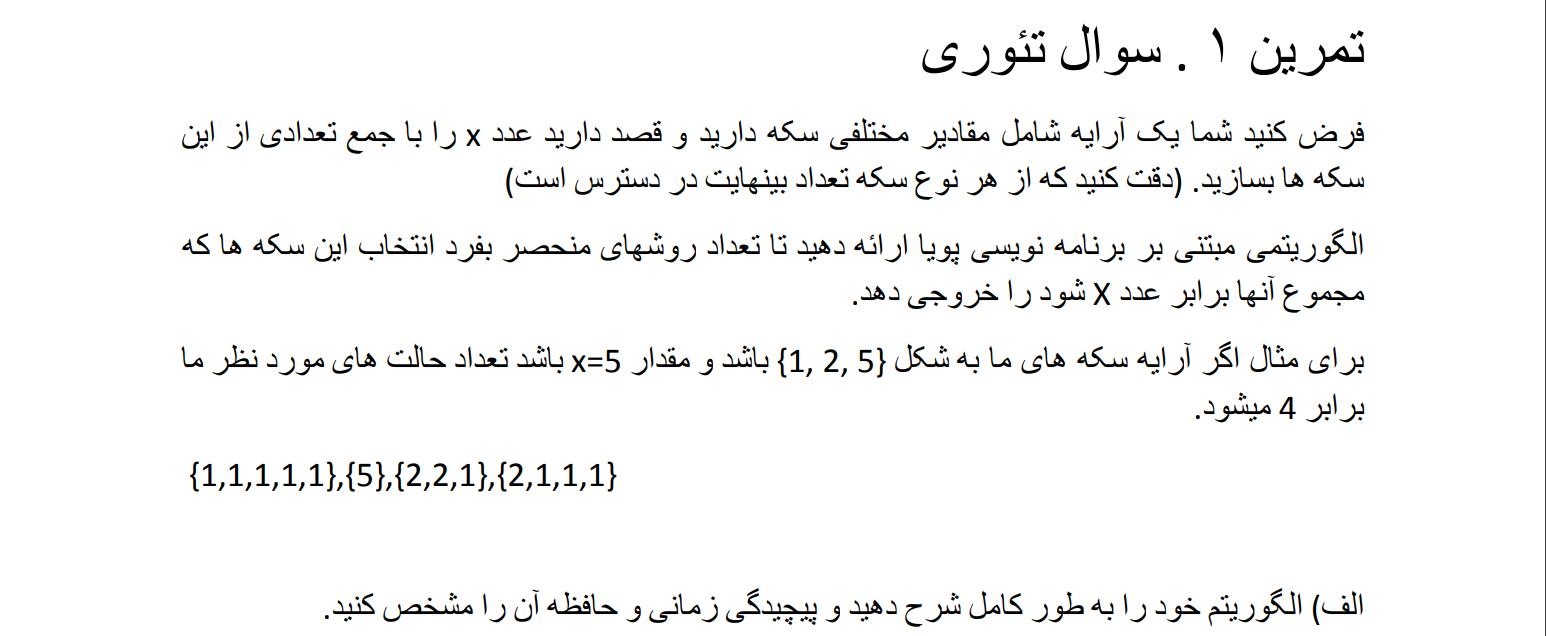
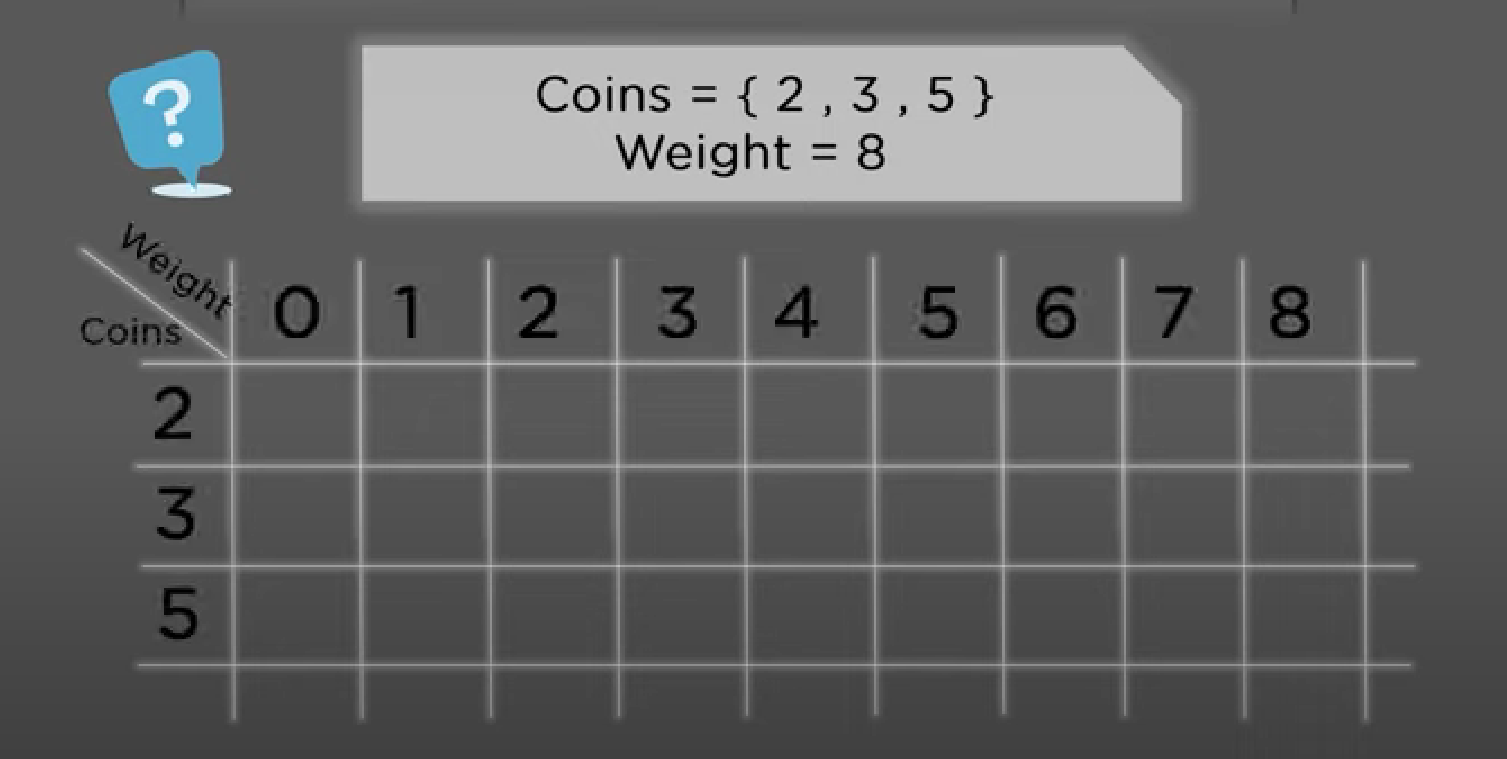
Amirali Farazmand

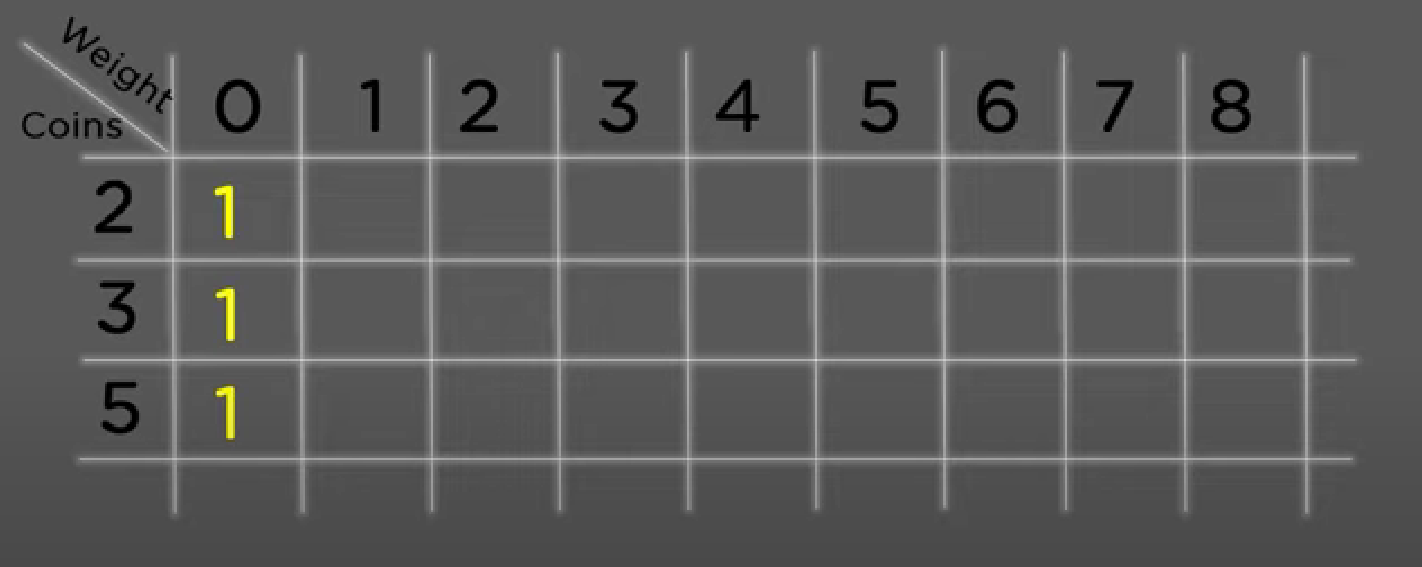
99522329

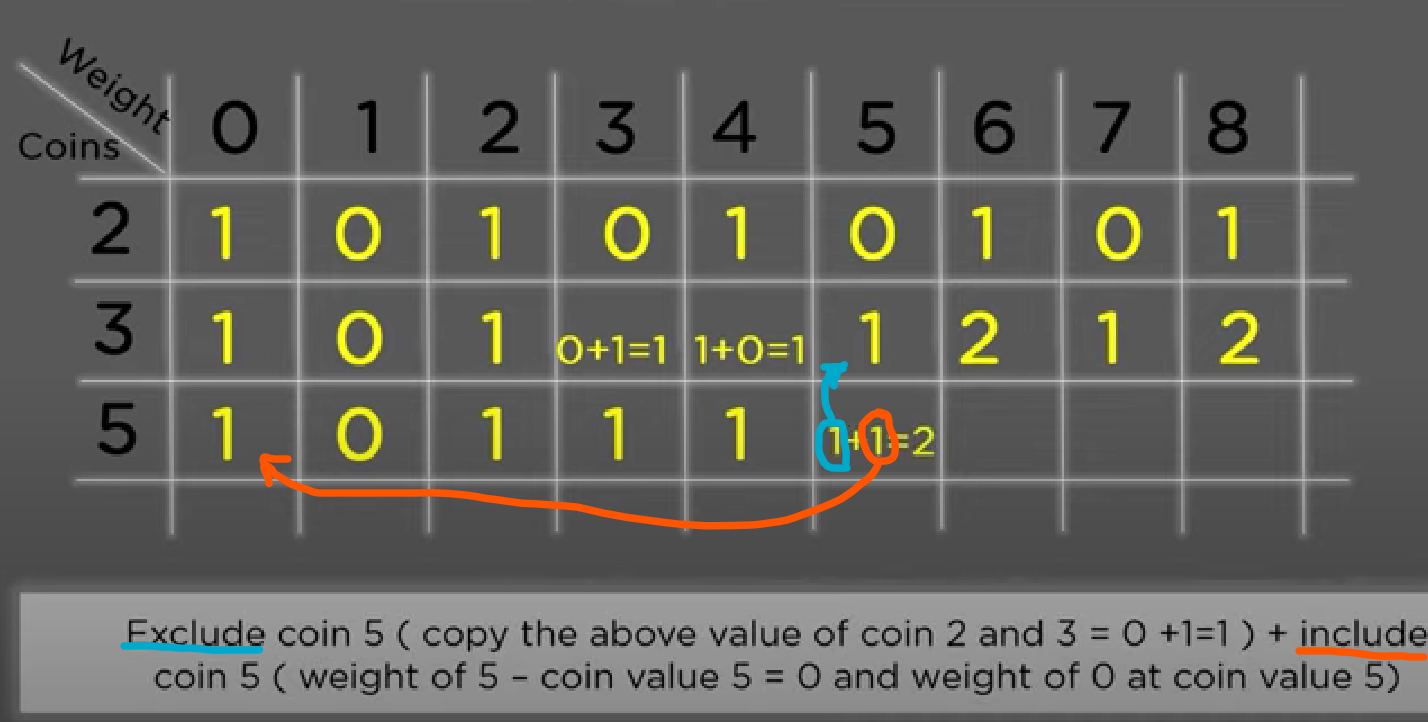


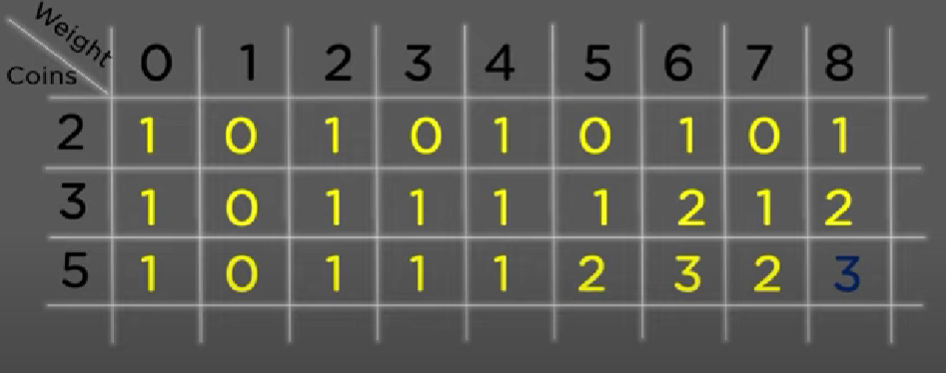
آرایه ی 2بعدی ای درست میکنیم که تعداد ستون هایش x+1 (0 تا x)و تعداد ردیف هایش هم تعداد سکه هایی که داریم میگذاریم.



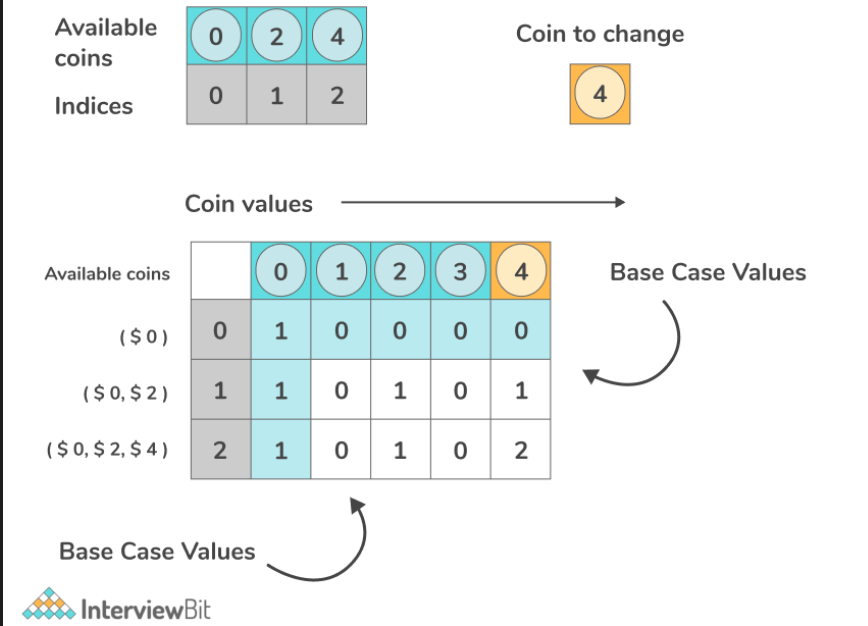
ستون اول که مقدار 0 باید با سکه ها پر شود را با 1 پر میکنیم چون یک حالت بیشتر ندارد.



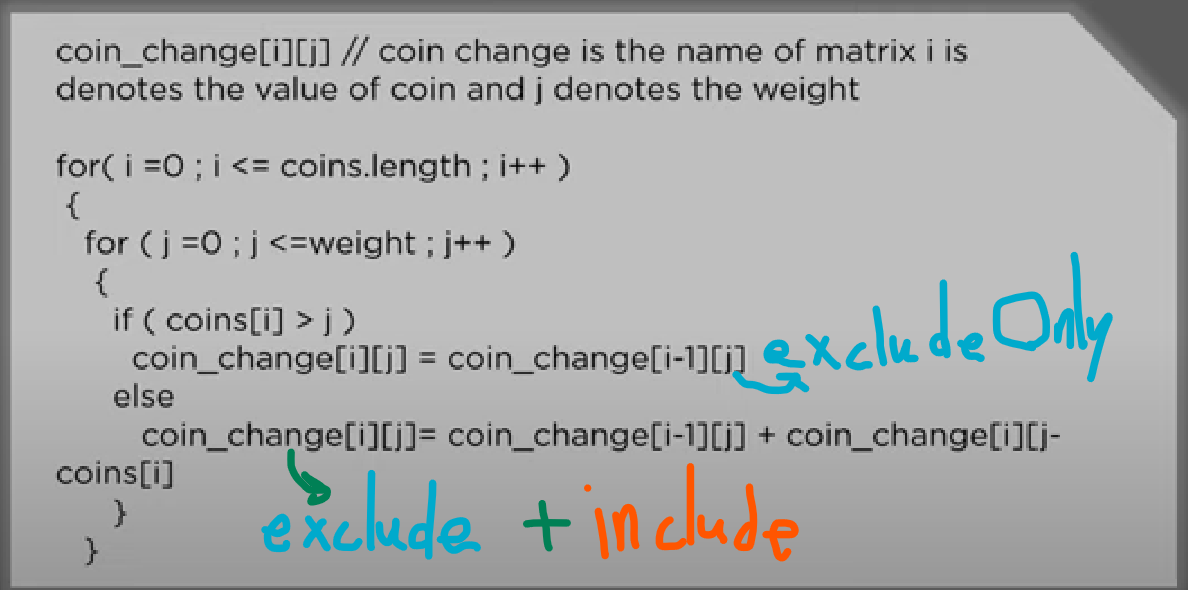
برای هر خانه(مثلا coin[x]) باید حساب کنیم مجموع حالاتی که آن سکه باشد(باقی مانده I – coin[x] که میشود خانه ای در همان ردیف اما با ستونی متفاوت) + آن سکه نباشد(که میشود خانه ی بالایی اش). این عمل را از بالای جدول شروع میکنیم و ردیف به ردیف پر میکنیم. خانه یی که اخرین weightو coin است جواب مسئله ی ما است.(به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



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 دردسر کمتری داریم در کد زدنش.

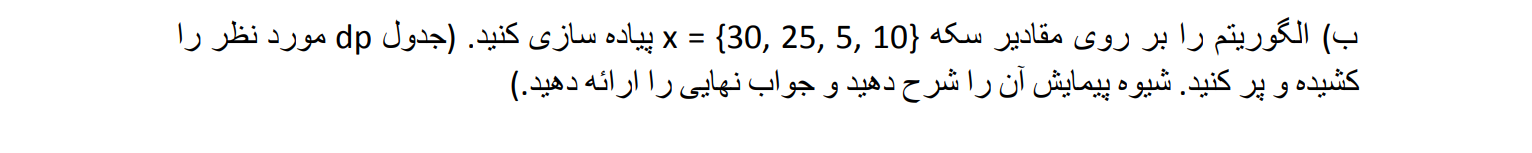
مثال دیگر: 

سودوکد:

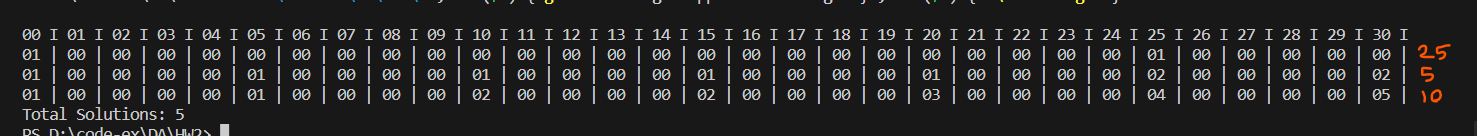


Time complexity: O(numberOfCoins \* TotalAmount)

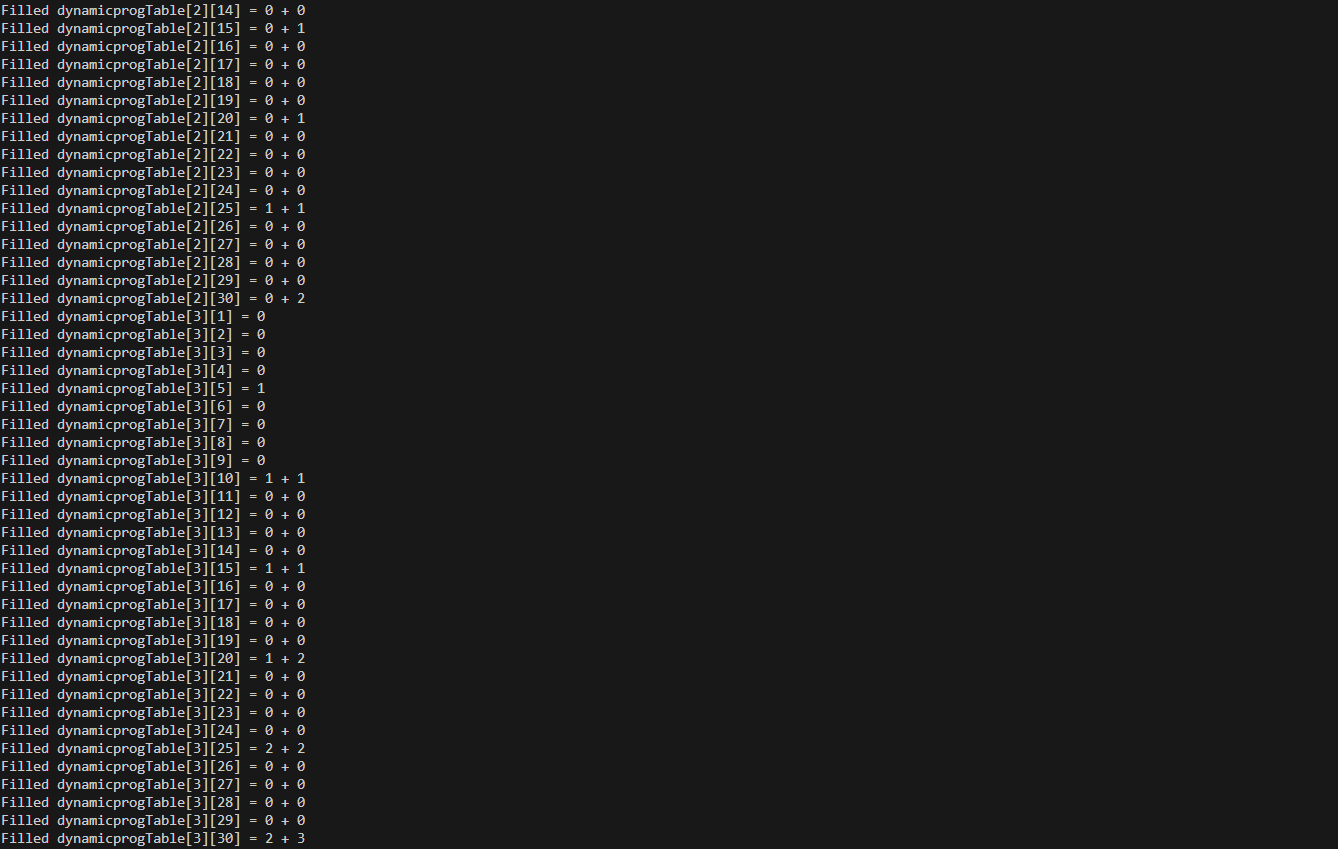
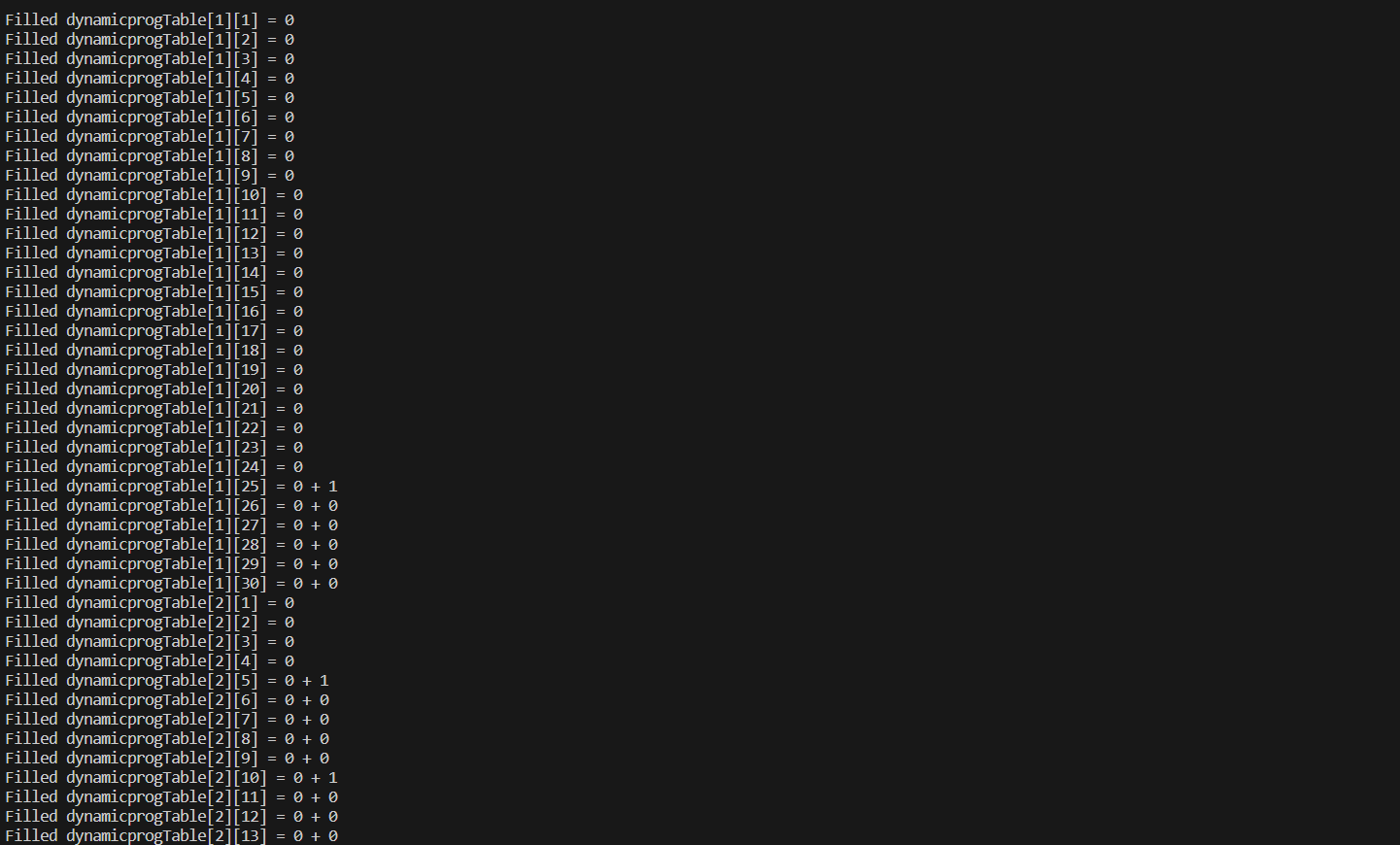
Space complexity: O(numberOfCoins \* TotalAmount)



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



نحوه پر شدن جدول:



[ویدیو توضیح الگوریتم](https://www.youtube.com/watch?v=WNv6T1uez6U)

[کد الگوریتم و توضیحات بیشتر](https://www.simplilearn.com/tutorials/data-structure-tutorial/coin-change-problem-with-dynamic-programming#:~:text=coin%20change%20problem.-,Introduction%20to%20Coin%20Change%20Problem,be%20constructed%2C%20return%20%2D1.)