

## HARDIK KHARE | 70765344

### HW 8.1 - Paying in Coins (25 pts)

You have a pocket full of coins (pennies, nickels, dimes, and quarters with values of 1, 5, 10, and 25 cents respectively) that you wish to get rid of. Each time you go to buy an item, you try to **pay with the most number of coins possible**.

Construct an algorithm that, given a cost and the number of coins you have available, pays the cost with as many coins as possible.

Your main will start with a pocket full of coins (loaded from stdin). Everytime an item is bought, the value of the item will be entered via stdin and your algorithm should be used to determine how you will pay for the item, and determine your remaining coins.

Input:

- starting number of each coin
- value of items to be bought

Output:

- for each item bought, the coins that were used, and the number of coins remaining in your pocket

### Section 1: Successful compilation of program

```
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % javac -Xlint COINSQ1.java  
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % java COINSQ1
```

### Section 2: program running on the provided example from the assignment

```
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % java COINSQ1  
Enter initial denominations  
7 10 1 3  
Keep entering value of each item(or -1 to quit)  
  
36 15 40 15 -1  
7 10 1 3  
6 6 0 0  
-----  
1 4 1 3  
0 3 0 0  
-----  
1 1 1 3  
0 1 1 1  
-----  
1 0 0 2  
Unable to pay!  
-----  
hardikkhare@Hardiks-MacBook-Pro AP HW 8 %
```

### HW 8.3 - DNA Sequence Alignment Part I (25 pts)

In DNA sequences, there are four types of repeating units (called bases) denoted as A, T, C, and G. Two opposite strands are aligned if All A's in one string are matched with T's in the opposing string, and all C's in one string are matched with G's in the opposing string.

An example of a valid alignment is:

```
A C C T C G
T G G A G C
```

...and an example of an invalid alignment is:

```
A G C T T G
T G C G A A
```

Assume that given opposing strands that do not align, and that you are allowed to put spacers to align them. For example:

We could transform this:

```
A G C T T G
T G C G A A
```

...into this:

```
A _ G C T T G
T G C G A A _
```

Write a program that takes two opposing strands of DNA, and determines the best way to align the sequences by inserting the minimum number of spacers.

#### Section 1: Successful compilation of program

```
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % javac -Xlint DNAQ1.java
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % java DNAQ1
```

#### Section 2: program running on the provided example from the assignment

```
[hardikkhare@Hardiks-MacBook-Pro AP HW 8 % java DNAQ1
Enter DNA strand 1
AGCTTG
Enter DNA strand 2
TGCGAA
Aligned Strands are:
A_GCTTG
AGCTTG_
hardikkhare@Hardiks-MacBook-Pro AP HW 8 % █
```

<https://leetcode.com/problems/coin-change/>

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

LeetCode Coding Challenge + GIVEAWAY!

+

🔔

👤

Coin Change

Submission Detail

188 / 188 test cases passed.  
Runtime: 56 ms  
Memory Usage: 44.1 MB

harry3997

My List

My Playground

Notebook

Submissions

Sessions

Progress

Points

Subscription

Orders

Sign out

Accepted Solutions Runtime Distribution

Zoom area by dragging across this chart

Accepted Solutions Memory Distribution

Zoom area by dragging across this chart

Invite friends to challenge Coin Change

f

🐦

👤

+

15

Submitted Code: 1 week, 1 day ago

Language: java

Edit Code

```
1 class Solution {
2     Integer[] dp;
3     public int helper(int[] coins, int amt){
4         if(amt==0) return 0;
5         if(dp[amt]==null){
6             int v=Integer.MAX_VALUE;
7             for(int i=0;i<coins.length;i++){
8                 if(amt-coins[i]>=0)
9                     v=Math.min(v,helper(coins,amt-coins[i]));
10            }
11            if(v!=Integer.MAX_VALUE) v++;
12            dp[amt]=v;
13        }
14        return dp[amt];
15    }
16    public int coinChange(int[] coins, int amount) {
17        dp=new Integer[amount+1];
18        int ans =helper(coins,amount);
19        return (ans==Integer.MAX_VALUE)?-1:ans;
20    }
21 }
```

<https://leetcode.com/problems/coin-change-2/>

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

LeetCode Coding Challenge + GIVEAWAY!

Coin Change 2

Submission Detail

28 / 28 test cases passed.  
Runtime: 22 ms  
Memory Usage: 43.8 MB

Accepted Solutions Runtime Distribution

Accepted Solutions Memory Distribution

Invited friends to challenge Coin Change 2

Submitted Code: 49 minutes ago

Language: java

Edit Code

```
1 class Solution {
2     Integer[][] dp;
3     public int helper(int[] coins, int ind, int amt){
4         if(amt==0) return 1;
5         if(amt<0||ind==coins.Length) return 0;
6         if(dp[ind][amt]==null)
7             dp[ind][amt]=helper(coins,ind+1,amt)+helper(coins,ind,amt-coins[ind]);
8
9         return dp[ind][amt];
10    }
11
12    public int change(int amount, int[] coins) {
13        dp=new Integer[coins.Length][amount+1];
14        return helper(coins,0,amount);
15    }
16 }
```

harry3997

My List

My Playground

Notebook

Submissions

Sessions

Progress

Points

Subscription

Orders

Sign out

30

25

20

15

10

5

0

0

50

100

150

200

250

300

Runtime (ms)

30

20

10

0

0

50

100

150

200

250

300

Zoom area by dragging across this chart

8

6

4

2

0

36000

38000

40000

42000

44000

46000

48000

50000

52000

54000

56000

Memory (KB)

8

6

4

2

0

37500

40000

42500

45000

47500

50000

52500

55000

57500

Zoom area by dragging across this chart

Facebook

Twitter

Reddit

Discord

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16