



PRACTICE

CERTIFICATION

COMPETE

JOBS

LEADERBOARD

Search



hackerrank682

Practice > Algorithms > Dynamic Programming > The Coin Change Problem > Editorial

The Coin Change Problem ☆

897.7 more points to get your next star!

Rank: 75298 | Points: 1302.3/2200



Problem

Submissions

Leaderboard

Discussions

Editorial

This is a classic dynamic programming problem that we can solve recursively.

Let's define a function named ***solve(i, make)*** that finds the number of ways to make ***make*** using coins from ***i*** to ***numCoins***. From this state, you have two options:

1. Take coin ***i***, making the next state ***solve(i, make - c[i])***.
2. Go on to the next coin, making the next state ***solve(i + 1, make)***.

The answer for ***solve(i, make)*** will be the sum of those two subproblems. If ***make*** becomes **0**, then return **1** (because there is only **1** way to make change for **0** units — don't take any coins!).

Don't forget to eliminate the overlapping subproblems using memoization, or your solution will time out. It's also important to note that the answer may be larger than a **32**-bit integer.

```
#include <bits/stdc++.h>

using namespace std;

int c[52];
int numCoins;
long long table[52][252];
bool calculated[52][252];
long long solve(int i, int make)
{
    if(make<0) return 0;
    if(make==0) return 1;
    if(i > numCoins) return 0;
    if(calculated[i][make] == false){ //eliminating overlap
        table[i][make] = solve(i, make - c[i]) + solve(i+1, make);
        calculated[i][make] = true;
    }
    return table[i][make];
}

int main(){
    int make;
    cin>>make>>numCoins;
    for(int i=1;i<=numCoins;i++){
        cin>>c[i];
    }
    cout<<solve(1, make)<<endl;
    return 0;
}
```

JP Tested by [John Pierce](#)

Problem Tester's code:

```
#!/bin/python3

import sys

def getWays(n, c):
    # Complete this function
    n_perms = [1]+[0]*n
    for coin in c:
        for i in range(coin, n+1):
            n_perms[i] += n_perms[i-coin]
    return n_perms[n]

n, m = input().strip().split(' ')
n, m = [int(n), int(m)]
c = list(map(int, input().strip().split(' ')))

ways = getWays(n, c)
print(ways)
```

STATISTICS

| | |
|------------------------------|--------------------------------|
| Difficulty: | Medium |
| Time Complexity: | O(N*M) |
| Required Knowledge: | dynamic programming, recursion |
| Publish Date: | Dec 01 2014 |
| This is a Practice Challenge | |

NEED HELP?

[View discussions](#)[View top submissions](#)

Feedback

Was this editorial helpful?

Yes

No

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)