

GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

Count number of ways to reach a given score in a game

Consider a game where a player can score 3 or 5 or 10 points in a move. Given a total score n, find number of ways to reach the given score.

Examples:

Input: n = 20

Output: 4

There are following 4 ways to reach 20

(10, 10)

(5, 5, 10)

(5, 5, 5, 5)

(3, 3, 3, 3, 3, 5)

Input: n = 13

Output: 2

There are following 2 ways to reach 13

(3, 5, 5)

(3, 10)

We strongly recommend you to minimize the browser and try this yourself first.

This problem is a variation of [coin change problem](#) and can be solved in $O(n)$ time and $O(n)$ auxiliary space.

The idea is to create a table of size $n+1$ to store counts of all scores from 0 to n . For every possible move (3, 5 and 10), increment values in table.

```
// A C program to count number of possible ways to a given score
// can be reached in a game where a move can earn 3 or 5 or 10
#include <stdio.h>
```

```
// Returns number of ways to reach score n
```

```
int count(int n)
```

```
{
```

```
    // table[i] will store count of solutions for
    // value i.
```

```
    int table[n+1], i;
```

```
    // Initialize all table values as 0
```

```
    memset(table, 0, sizeof(table));
```

```
    // Base case (If given value is 0)
```

```
    table[0] = 1;
```

```
    // One by one consider given 3 moves and update the table[]
```

```
    // values after the index greater than or equal to the
```

```
    // value of the picked move
```

```
    for (i=3; i<=n; i++)
```

```
        table[i] += table[i-3];
```

```
    for (i=5; i<=n; i++)
```

```
        table[i] += table[i-5];
```

```
    for (i=10; i<=n; i++)
```

```
        table[i] += table[i-10];
```

```
    return table[n];
```

```
}
```

```
// Driver program
```

```
int main(void)
```

```
{
```

```
    int n = 20;
```

```
    printf("Count for %d is %d\n", n, count(n));
```

```
    n = 13;
```

```
    printf("Count for %d is %d", n, count(n));
```

```
    return 0;
```

}

Output:

Count for 20 is 4
 Count for 13 is 2

Exercise: How to count score when (10, 5, 5), (5, 5, 10) and (5, 10, 5) are considered as different sequences of moves. Similarly, (5, 3, 3), (3, 5, 3) and (3, 3, 5) are considered different.

This article is contributed by **Rajeev Arora**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- [Linearity of Expectation](#)
- [Iterative Tower of Hanoi](#)
- [Count possible ways to construct buildings](#)
- [Build Lowest Number by Removing n digits from a given number](#)
- [Set Cover Problem | Set 1 \(Greedy Approximate Algorithm\)](#)
- [Find number of days between two given dates](#)
- [How to print maximum number of A's using given four keys](#)
- [Write an iterative O\(Log y\) function for pow\(x, y\)](#)

Tags: [Dynamic Programming](#)



Tweet

< 3

+1

< 1

Writing code in comment? Please use ideone.com and share the link here.

28 Comments

GeeksforGeeks



Login ▾

Recommend

Share

Sort by Newest ▾



Join the discussion...



Guest • 25 days ago

public class Point{

int x,y;

Point(){x=0;y=0;}

}

Enum Direction{E,N,W,S};

public class DirectRobot{

Point point;

DirectRobot(){

}

public static void main(String args[]){

~~point=new Point();~~

[see more](#)

^ | v • Reply • Share ›



creeping_death • 2 months ago

Ruby solution $O(N \cdot C)$ where N is target and C is acceptable / valid scores:

<https://gist.githubusercontent...>

^ | v • Reply • Share ›



Sourabh Jain • 2 months ago

i think we can represent this as

$$x \cdot 3 + y \cdot 5 + z \cdot 10 = n;$$

$$x \cdot 3 + 5 \cdot (y + 2z) = n;$$

$$(y + 2z) = (n - x \cdot 3) / 5;$$

total no of possible soln for $(y + 2z) = m$ are $(m/2 + 1)$ from intuition , and we can also prove it.

so now we can iterate x from 0 to n and check $(n - x \cdot 3) \% 5 == 0$ and find soln for $(y + 2z)$;

Time Complexity $O(n)$

Space Complexity $O(1)$

Plz correct me if i am wrong

^ | v • Reply • Share ›



The Internet • 2 months ago

Here is a shorter version: <http://ideone.com/qRHqbc>

^ | v • Reply • Share ›



dd • 2 months ago

SAPOff Campus Hiring March 2015 <http://www.gohired.in/2015/03/...>

^ | v • Reply • Share ›



SANKHYA • 2 months ago

RECURSION:::

```
#include <iostream>

using namespace std;

void now(int score, int *result){

if(score < 3)

return;

if(score == 3 || score == 5 || score == 10){

*result = *result + 1;

}

now(score-3, result);

now(score-5, result);
```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) ›**sk** • 2 months ago

For exercise problem we can use this recurrence relation

$$T(n) = T(n-3) + T(n-5) + T(n-10)$$

$$T(0) = T(1) = T(2) = T(4) = T(7) = 0$$

$$T(3) = T(5) = T(6) = T(9) = 1$$

$$T(8) = T(10) = 2$$

Now use memoization technique to compute $n > 10$.

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Satu0King** • 2 months ago

```
#include<iostream>
```

```
using namespace std;
```

```
int pos (int num)
```

```
{
```

```
int count=0;
```

```
for(int i=0; i<=num/10; i++)
```

```
for (int j=0; j<=(num-10*i)/5; j++)
```

```

if((num-10*i-5*j)%3==0)count++;

return count;

}

int main()

{

cout <<pos(13); }="">
3 ^ | v • Reply • Share ›

```



sachin • 2 months ago

The solution to the exercise:

```

#include<stdio.h>

static int i=0;

static int cont=0;

void fun(int a[2],int n,int x1)

{

int z=1;

int sum=0;

int x=a[i];

i++;

while(sum<=n)

```

[see more](#)

^ | v • Reply • Share ›



Guest • 2 months ago

Solution to the exercise,Kindly check if anything is incorrect:

```

#include<stdio.h>

static int i=0;

static int cont=0;

void fun(int a[2],int n,int x1)

{

int z=1;

```

```

int sum=0;
int x=a[i];
i++;
while(sum<=n)
{
sum=sum+x;
if(sum==n)
{
cont++;
}}
int g;
for(g=0;g<x1;g++) {="" sum="0;" z="1;" int="" f;="" if(a[g]!="x)" {="" while(sum<="n)" {=""
sum="0;" f="z*x;" sum="sum+f+a[g];" z++;="" if(sum=="n)" {="" cont++;="" }}}="" int=""
main()="" {="" int="" x1,y;="" int="" n="13;" int="" a[3]="{3,5,10};" x1="sizeof(a)/sizeof(a[0]);"
for(y="0;y<=x1;y++)" {="" fun(a,n,x1);="" }="" printf("%d\n",cont);="" }="">

```

^ | v • Reply • Share ›



Nikhil Jindal • 2 months ago

```
#include <iostream>
```

```
int counter=0;
```

```
using namespace std;
```

```
void three(int n)
```

```
{
```

```
int a;
```

```
a=n/3;
```

```
for(int i=a;i>0;i--)
```

```
{
```

```
int b=n-(3*i);
```

```
if(b==0)
```

[see more](#)

^ | v • Reply • Share ›



Sarath S Pillai • 3 months ago

python code with memoization

```
possible_steps = sorted([3,5,10])
```

```
def memoize(f):
    """ Memoization decorator for a function taking a single argument """
    class memodict(dict):
        """ The dictionary that will act as hashmap for Memoization """
        def __missing__(self, key):
            ret = self[key] = f(key)
            return ret
    return memodict().__getitem__

@memoize
def number_of_ways(score):
    if score == 0:
        return 1
    ways = 0;
    for i in possible_steps:
        if i <= score:
            ways += number_of_ways(score - i)
        else:
            break
    return ways

print number_of_ways(13)
```

1 ^ | v • Reply • Share ›



Ankit Goyal • 3 months ago

For the exercise ---

Recursive Algo :

```
int Count(int n)
{
    if (n < 3) return 0;
    if (n == 3 || n == 5) return 1;
    if (n == 10) return 2;

    return Count(n-3) + Count(n-5) + Count(n-10);
}
```

We can easily use memoization here to get the $O(n)$ time complexity.

3 ^ | v • Reply • Share ›



Ash ➔ Ankit Goyal • 2 months ago

this will treat 5,10,5 and 5,5,10 as different combination

1 ^ | v • Reply • Share ›

**Ankit Goyal** → Ash • 2 months ago

That's what the exercise problem asks for right?

^ | v • Reply • Share ›

**Shubham Sharma** → Ash • 2 months ago

Can we subtract the solutions i.e
 return count(n-3)+count(n-5)+count(n-10)-count(num-8)-count(num-15)-
 count(num-13) +count(num-18);

^ | v • Reply • Share ›

**Ankit Goyal** • 3 months ago

I think there is an explicit constraint on the output which has been kept implicit.
 It should have been mentioned that sequences with the same elements should be counted as 1 solution only because sequences with same elements but different order are also a part of the solution, intuitively and that's what the exercise problem also is all about.

So, to say, in the coin change problem, its specifically mentioned that "the order of coins doesn't matter"

^ | v • Reply • Share ›

**tushar** • 3 months ago

Exercise: How to count score when (10, 5, 5), (5, 5, 10) and (5, 10, 5) are considered as different sequences of moves. Similarly, (5, 3, 3), (3, 5, 3) and (3, 3, 5) are considered different.

SOLUTION:

```
int count(int n)
{
  int table[n+1], i;
  memset(table, 0, sizeof(table));
  table[0] = 1;
  for (i=3; i<=n; i++)
  {
    if(i>=3)
      table[i] += table[i-3];
    if(i>=5)
      table[i] += table[i-5];
    if(i>=10)
      table[i] += table[i-10];
  }
  return table[n];
}
```

^ | v • Reply • Share ›

**Mind Boggler** • 3 months ago

This problem can also be viewed as variation to staircase problem?

^ | v • Reply • Share ›

**khan** • 3 months ago

pl provide recursive solution also

^ | v • Reply • Share ›

**sid** • 3 months ago

```
int numWays(int n)
```

```
//
```

```
int num[] <- {3, 5, 10}
```

```
int nWays <- 0
```

```
for i <- 0 to n, i <- i + num[0]
```

```
do
```

```
j <- 0
```

```
k <- n - n mod num[2]
```

```
nSum <- n - i
```

```
while j is less than or equal to n and k is greater than or equal to 0
```

```
do
```

```
nTemp <- j + k
```

```
if nTemp is equal to nSum
```

```
then
```

```
nWays++
```

```
j <- j + num[1]
```

```
k <- k - num[2]
```

```
else if nTemp is less than nSum
```

see more

^ | v • Reply • Share ›

**Dipankar Jana** • 3 months ago

Please provide recursive solution as well.

^ | v • Reply • Share ›

**kha** • 3 months ago

exercise????

^ | v • Reply • Share ›

**Ravi Ojha** • 3 months ago

Generally, the complexity is actually $O(\text{types_of_coins} * n)$. I was surprised when I read $O(n)$ but figured that it runs $O(n)$ for each type of coin from the code. If we have lot of coin denominations then we can't say that it's $O(n)$.

^ | v • Reply • Share ›



Miguel Oliveira → Ravi Ojha · 3 months ago

There is a constant number of coins in this particular problem. Hence the constant is ignored in big-O notation

^ | v · Reply · Share ›



Ravi Ojha → Miguel Oliveira · 3 months ago

Yup.. I know. Edited my comment to add generality. Thanks.

^ | v · Reply · Share ›



Meraj Ahmed · 3 months ago

In example 1 where $n = 20$, the second instance is wrongly printed "(5, 10, 10)", it should be "(5, 5, 10)".

1 ^ | v · Reply · Share ›



GeeksforGeeks Mod → Meraj Ahmed · 3 months ago

Thanks for pointing this out. We have corrected the example.

^ | v · Reply · Share ›



Subscribe



Add Disqus to your site



Privacy

-
-
-
-

- [Interview Experiences](#)
- [Advanced Data Structures](#)
- [Dynamic Programming](#)
- [Greedy Algorithms](#)
- [Backtracking](#)
- [Pattern Searching](#)
- [Divide & Conquer](#)
- [Mathematical Algorithms](#)
- [Recursion](#)

- [Geometric Algorithms](#)

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST.](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

• Recent Comments

- [It_k](#)

i need help for coding this function in java...

[Java Programming Language](#) · [2 hours ago](#)

- [Piyush](#)

What is the purpose of else if (recStack[*i])...

[Detect Cycle in a Directed Graph](#) · [2 hours ago](#)

- [Andy Toh](#)

My compile-time solution, which agrees with the...

[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [2 hours ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [2 hours ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [3 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [3 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team