

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 Possible Decodings of a given Digit Sequence

Let 1 represent 'A', 2 represents 'B', etc. Given a digit sequence, count the number of possible decodings of the given digit sequence.

Examples:

Input: digits[] = "121"

Output: 3

// The possible decodings are "ABA", "AU", "LA"

Input: digits[] = "1234"

Output: 3

// The possible decodings are "ABCD", "LCD", "AWD"

An empty digit sequence is considered to have one decoding. It may be assumed that the input contains valid digits from 0 to 9 and there are no leading 0's, no extra trailing 0's and no two or more consecutive 0's.

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

This problem is recursive and can be broken in sub-problems. We start from end of the given digit sequence. We initialize the total count of decodings as 0. We recur for two subproblems.

- 1) If the last digit is non-zero, recur for remaining (n-1) digits and add the result to total count.
- 2) If the last two digits form a valid character (or smaller than 27), recur for remaining (n-2) digits and add the result to total count.

Following is C++ implementation of the above approach.

```
// A naive recursive C++ implementation to count number of decodings
// that can be formed from a given digit sequence
#include <iostream>
#include <cstring>
using namespace std;

// Given a digit sequence of length n, returns count of possible
// decodings by replacing 1 with A, 2 with B, ... 26 with Z
int countDecoding(char *digits, int n)
{
    // base cases
    if (n == 0 || n == 1)
        return 1;

    int count = 0; // Initialize count

    // If the last digit is not 0, then last digit must add to
    // the number of words
    if (digits[n-1] > '0')
        count = countDecoding(digits, n-1);

    // If the last two digits form a number smaller than or equal to 26,
    // then consider last two digits and recur
    if (digits[n-2] < '2' || (digits[n-2] == '2' && digits[n-1] < '7'))
        count += countDecoding(digits, n-2);

    return count;
}

// Driver program to test above function
int main()
{
    char digits[] = "1234";
    int n = strlen(digits);
    cout << "Count is " << countDecoding(digits, n);
    return 0;
}
```

Output:

Count is 3

The time complexity of above the code is exponential. If we take a closer look at the above program, we can observe that the recursive solution is similar to [Fibonacci Numbers](#). Therefore, we can optimize the above solution to work in $O(n)$ time using [Dynamic Programming](#). Following is C++ implementation for the same.

```
// A Dynamic Programming based C++ implementation to count decodings
#include <iostream>
#include <cstring>
using namespace std;

// A Dynamic Programming based function to count decodings
int countDecodingDP(char *digits, int n)
{
    int count[n+1]; // A table to store results of subproblems
    count[0] = 1;
    count[1] = 1;

    for (int i = 2; i <= n; i++)
    {
        count[i] = 0;

        // If the last digit is not 0, then last digit must add to
        // the number of words
        if (digits[i-1] > '0')
            count[i] = count[i-1];

        // If second last digit is smaller than 2 and last digit is
        // smaller than 7, then last two digits form a valid character
        if (digits[i-2] < '2' || (digits[i-2] == '2' && digits[i-1] < '7'))
            count[i] += count[i-2];
    }
    return count[n];
}

// Driver program to test above function
int main()
{
    char digits[] = "1234";
    int n = strlen(digits);
    cout << "Count is " << countDecodingDP(digits, n);
    return 0;
}
```

Output:

Count is 3

Time Complexity of the above solution is $O(n)$ and it requires $O(n)$ auxiliary space. We can reduce auxiliary space to $O(1)$ by using space optimized version discussed in the [Fibonacci Number Post](#).

Please write comments if you find anything incorrect, or you want to share more information about the

topic discussed above

Related Topics:

- [Recursively print all sentences that can be formed from list of word lists](#)
- [Check if a given sequence of moves for a robot is circular or not](#)
- [Find the longest substring with k unique characters in a given string](#)
- [Function to find Number of customers who could not get a computer](#)
- [Find maximum depth of nested parenthesis in a string](#)
- [Find all distinct palindromic sub-strings of a given string](#)
- [Find if a given string can be represented from a substring by iterating the substring “n” times](#)
- [Suffix Tree Application 6 – Longest Palindromic Substring](#)

Tags: [Dynamic Programming](#), [Fibonacci numbers](#), [MathematicalAlgo](#)



Tweet

3

+1

2

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

76 Comments

GeeksforGeeks

1

Login ▾

♥ Recommend 1

🔗 Share

Sort by Newest ▾



Join the discussion...



Saurabh • 25 days ago

Th

^ | v • Reply • Share ›



dd • 2 months ago

understand method here : <https://www.youtube.com/watch?...>

^ | v • Reply • Share ›



Sudeep Soni • 2 months ago

In java: <http://ideone.com/V2L93m>

^ | v • Reply • Share ›



codeStorm3 • 3 months ago

'C' Code to print Possible Decodings of a given Digit Sequence !!!

<http://ideone.com/uPMFzJ>

If find any problem in code please comment?

^ | v • Reply • Share ›



neer1304 • 4 months ago



O(1) space solution

<http://ideone.com/5hQNtU>

^ | v • Reply • Share ›



Itachi Uchiha → neer1304 • 2 months ago

Gives incorrect output!

^ | v • Reply • Share ›



neer1304 → Itachi Uchiha • 2 months ago

Thanks for the input. Kindly see the updated code below:-

<http://ideone.com/OoK8IB>

Let me know if you still face some issue.

^ | v • Reply • Share ›



manish • 4 months ago

<http://chopapp.com/#copu3mq9>

Pasted the code at above URL

^ | v • Reply • Share ›



manish • 4 months ago

I have got more straight forward algorithm but. You just need to check if the last to digits of the numbers are in between 11 and 26(inclusive). If so, it adds a count. Simple:

```
def digicount(digit):
    count = 1
    while len(digit) > 1:
        x = digit[len(digit) - 2:]
        if int(int(x) in xrange(11, 27)):
            count += 1;
        digit = digit[:len(digit) - 1]
    return count
```

```
if __name__ == "__main__":
    print digicount("1201")
```

^ | v • Reply • Share ›



Mrigank Dembla • 6 months ago

@GeeksforGeeks The above code will give wrong answers for string like 101,11101.

Condition should be modified to :-

```
if (digits[i-2] > '0' && (digits[i-2] < '2' || digits[i-2]=='2' && digits[i-1] < '7') )
```

6 ^ | v • Reply • Share ›

**Anshul** → Mrigank Dembla • 5 months ago

thanks mate tht was good oservation.

```

public static int numOfDecodedWords(String code){
    int num = 0;
    if(code.length() == 0){
        return 0;
    }
    if(code.length() == 1){
        int firstChar = Integer.parseInt(code.substring(0 , 1));
        if(firstChar == 0)
            return -1;
        else
            return 1;
    }
    if(code.length() == 1){
        return 1;
    }
    if(code.length() == 2){

```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) ›**sk** • 6 months ago

@GeeksforGeeks

There is a bug in 2nd if condition in dp solution .
 It should be if(d[i-2]=='1' || (d[i-2]=='2' && d[i-1]<'7'))

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Akshit Bhatia** • 6 months ago

try it out

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Akshit Bhatia** • 6 months ago

a c language implementation inspired by geeksforgeeks but with little modification

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
static int count=1;
void decode(char* digit,int n){
    int i=0;
    for(i=n;i>=0;i--){
        if(digit[i-1]<'2'&&digit[i-1]>'0' || digit[i-1]=='2'&&digit[i]<'7')

```

```

{ count++; continue;}
else continue;

}}

void main(){
char* digit="12340";
int n=strlen(digit)-1;
int counts=0;
clrscr();
decode(digit,n);
printf("%d",count);getch();}

```

^ | v • Reply • Share ›



Liger • 7 months ago

If the above code is including 0 as an input means 0 needs to be consider with the adjacent number then it is ok but if 0 means no alphabet then it is wrong.

1 ^ | v • Reply • Share ›



Ram • 7 months ago

The above code is wrong. Please check for the idput 12014. The code gives output as 4. But actual output is 5.

12014 = abad, atad, atn, lad, ld

2 ^ | v • Reply • Share ›



Akshit Bhatia → Ram • 6 months ago

@ram and @cyranosays how is ld possible
please do reply at bhatia_akshit@live.com also
thank you

^ | v • Reply • Share ›



cyranosays → Ram • 7 months ago

It is actually 6

12014 = abad, atad, atn, lad, ld, ln

^ | v • Reply • Share ›



Ram → Ram • 7 months ago

This is the flaw with the DP approach. I did not check for the Normal approach.

2 ^ | v • Reply • Share ›



veer verma → Ram • 7 months ago

I agree with you. There is a flaw in the optimized DP approach.

Flaw is in the statement:-

```
if (digits[i-2] < '2' || (digits[i-2] == '2' && digits[i-1] < '7'))
```

Here you are not checking that previous digit should not be 0.
So the correct statement would be:-

```
if ((digits[i-2] < '2' && digits[i-2] != '0') || (digits[i-2] == '2' && digits[i-1] < '7'))
```

^ | v • Reply • Share ›



Praveen Pandey • 7 months ago

I have solved this Problem in PHP, approach is similar, discussed in (Overlapping Subproblems Property).

It also handles 0 case if input given is valid.

Complexity: O(n)

Please do comments for improvements.

```
function countDecodingDp($numberString, $start, $end, &$amp;decodingCounts) {
```

```
$count = 0;
```

```
if ($end <= 1) {
```

```
    return 1;
```

```
}
```

```
$substring = substr($numberString, $start, $end);
```

```
// if already solved
```

```
if(isset($decodingCounts[$start.'_'.$end])) {
```

```
    return $decodingCounts[$start.'_'.$end];
```

[see more](#)

1 ^ | v • Reply • Share ›



Guest • 7 months ago

Here is my solution in PHP language.

complexity : O(n)

I have used DP approach similar to approached discussed in (Overlapping Subproblems Property) problem.

```
function countDecodingDp($numberString, $start, $end, &$amp;decodingCounts) {
```

```
$count = 0;
```

```
if ($end <= 1) {
```

```
    return 1;
```



```
}
```

```
$substring = substr($numberSting, $start, $end);
```

```
if(isset($decodingCounts[$start.'_'.$end])) {
```

[see more](#)

^ | v • Reply • Share ›



aa1992 • 7 months ago

here is the java code to print the decodings along with the count

<http://ideone.com/iyEKGM>

Please report if any error

1 ^ | v • Reply • Share ›



Praveen Pandey • 8 months ago

wrong code

^ | v • Reply • Share ›



Guest → Praveen Pandey • 7 months ago

Please also do mention the errors if you have really found any

^ | v • Reply • Share ›



aa1992 → Praveen Pandey • 7 months ago

why do you think the code is wrong?here is the java code to print all the decodings along with count.

<http://ideone.com/iyEKGM>

^ | v • Reply • Share ›



Aveek Biswas • 8 months ago

The code doesn't work for digits="12340"

The ouput is coming zero.

^ | v • Reply • Share ›



Vito → Aveek Biswas • 2 months ago

As per question, you cannot have leading or trailing zeroes.

^ | v • Reply • Share ›



pandu → Aveek Biswas • 8 months ago

output will be 0 only. Check again

^ | v • Reply • Share ›



Akshit Bhatia → pandu • 6 months ago



yeah the output must be 3
ain't it?

^ | v • Reply • Share ›



mohammad • 8 months ago

for 23 output should be 3...but your solution is giving 2.

^ | v • Reply • Share ›



Yiding Zhou → mohammad • 8 months ago

23 : 2,3 or 23. only 2?

1 ^ | v • Reply • Share ›



Akshit Bhatia → Yiding Zhou • 6 months ago

exactly only 2

^ | v • Reply • Share ›



Sandipan Manna • 8 months ago

It can be more space optimized, if we use only two variables (PrevPrev, Prev) instead of keeping a whole array.

^ | v • Reply • Share ›



Paparao Veeragandham • 9 months ago

My Own Method:

```
void PrintAllDecodings(String data[], char temp[], char results[], int index)
```

```
{
    int i , n= data.length();
    if( n ==0)
    { results[n] = '\0';
      printf("%s", results);
      return;
    }
    for( i = 0; i < n; i++)
    {
        if(FindIndex(data, 0 , i) < 26)
        {
            results[index] = temp[Findindex(data,0,i)];
            PrintAllDecodings(data.substr(0,i),temp,results,index+1);
        }
    }
}
```

^ | v • Reply • Share ›



prasun_goyal • 9 months ago

these two codes give wrong output for the input "1202"



these two codes give wrong output for the input 1202

The expected output is 1 ("ATB" is the only valid string)

but above codes give 3 as output.

The same question is present in problem set of SPOJ (Problem code- ACODE)

my ideone link for solution of ACODE : <http://ideone.com/Esc7n8>

2 ^ | v • Reply • Share ›



Aveek Biswas → prasun_goyal • 7 months ago

<http://ideone.com/YDJHIV>

^ | v • Reply • Share ›



pandu → prasun_goyal • 8 months ago

exactly

^ | v • Reply • Share ›



Ajay Sreeram • 9 months ago

count[5] is used only for up to count[7] then why to store n element values

this below code may reduce the usage of memory:

```
public static int rec(char digits[],int n)
{
    int a,b,c;
    a=b=c=1;
    for(int i=2;i<=n;i++){
        c=0;
        if(digits[i-1]>'0')
            c=b;
        if(digits[i-2]<'2'||(digits[i-2]=='2'&&digits[i-1]<'7'))
            c=c+a;
        a=b;
        b=c;
    }
    return c;
}
```

^ | v • Reply • Share ›



Gaurav Gupta • 9 months ago

complexity O(n); Outputs the decoded characters as well. Works for all cases

<http://ideone.com/U8MXMd>

^ | v • Reply • Share ›



Indranil Nandy • 10 months ago

This will fail for "101". Expected output is 1, while the above code snippet will give 2 as result.

The last condition should be modified as

```
digits[i-2] > '0' && ( digits[i-2] < '2' || (digits[i-2] == '2' && digits[i-1] < '7') )
```

Will appreciate if you take a note of this.

Thanks.

4 ^ | v • Reply • Share ›



Ram → Indranil Nandy • 7 months ago

It is 2 only... 101 = aa, ja.

^ | v • Reply • Share ›



vivek • 10 months ago

wrong for 1301

^ | v • Reply • Share ›



GeeksforGeeks Mod → vivek • 10 months ago

Could you please provide more details like what is expected and what did you get?

^ | v • Reply • Share ›



Kausik → GeeksforGeeks • 10 months ago

I think 1301 is not a valid sequence because then it has to be broken as 1|30|1 and 30 doesn't represent a valid symbol.

But according to me, still there is a problem. Take the example 1201. then the only valid coding will be ATB (1|20|1). The the solution is giving answer 3 (1|20|2, 1|2|02, 12|20).

I think the last if statement should be `if(digits[n-2]== '1' || (digits[n-2] == '2' && digits[n-1] < '7'))`

^ | v • Reply • Share ›



Anurag Singh → Kausik • 9 months ago

I think either ZERO should be given one decode character (e.g. 0 -> A, 1 -> B etc) or ZERO should not be part of a valid input at all. The problem statement says that ZERO can only appear in between, not at either end. If that's the case, then sub-problem will have ZERO at the end and that will make a invalid sub-problem. e.g. if input is 1201, then 120 will be a sub-problem and this is going to be a invalid sub-problem.

2 ^ | v • Reply • Share ›



Amit Kaushik • 10 months ago

more generic and compact code:

```
int countDec(char* seq){
```

```
int count = 0;

if(*seq=='\0' || *(seq+1)=='\0')

return 1;

if(*seq > '0')

count = countDec(seq+1);

if((*seq < '2' || (*seq=='2' && *(seq+1)<'7'))&&*seq>'0')

count+=countDec(seq+2);

return count;
}
```

1 ^ | v • Reply • Share ›



guest • 10 months ago

good dp problem

^ | v • Reply • Share ›



Guest • 10 months ago

For test case 12222 the solution is failing

^ | v • Reply • Share ›



GeeksforGeeks Mod ➔ **Guest** • 10 months ago

Could you please provide more details like what is expected and what did you get.

^ | v • Reply • Share ›

Load more comments

[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