Program for nth Catalan Number

Catalan numbers are a sequence of natural numbers that occurs in many interesting counting problems like following.

1) Count the number of expressions containing n pairs of parentheses which are correctly matched. For n = 3, possible expressions are ((())), ()(()), ()()(), (())(), (()()).

2) Count the number of possible Binary Search Trees with n keys (See this)

3) Count the number of full binary trees (A rooted binary tree is full if every vertex has either two children or no children) with n+1 leaves.

See this for more applications.

The first few Catalan numbers for n = 0, 1, 2, 3, … are **1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, …**

Recursive Solution
Catalan numbers satisfy the following recursive formula.
[Tex] C_0 = 1 \quad \mbox{and} \quad C_{n+1}=\sum_{i=0}^{n}C_i\,C_{n-i}\quad\text{for }n\ge 0; [/Tex]

Following is C++ implementation of above recursive formula.

```
#include<iostream>
using namespace std;

// A recursive function to find nth catalan number
unsigned long int catalan(unsigned int n)
{
    // Base case
    if (n <= 1) return 1;

    // catalan(n) is sum of catalan(i)*catalan(n-i-1)
    unsigned long int res = 0;
    for (int i=0; i<n; i++)
        res += catalan(i)*catalan(n-i-1);

    return res;
}

// Driver program to test above function
int main()
{
    for (int i=0; i<10; i++)
        cout << catalan(i) << " ";
    return 0;
}
```

Output :

```
1 1 2 5 14 42 132 429 1430 4862
```

Time complexity of above implementation is equivalent to nth catalan number.

[Tex] T(n) = \sum_{i=0}^{n-1} {T(i)*T(n-i)}\quad\text{for }n\ge 0;[/Tex]

The value of nth catalan number is exponential that makes the time complexity exponential.

Dynamic Programming Solution
We can observe that the above recursive implementation does a lot of repeated work (we can the same by drawing recursion tree). Since there are overlapping subproblems, we can use dynamic programming for this. Following is a Dynamic programming based implementation in C++.

```
#include<iostream>
using namespace std;
```

```
// A dynamic programming based function to find nth
// Catalan number
unsigned long int catalanDP(unsigned int n)
{
    // Table to store results of subproblems
    unsigned long int catalan[n+1];

    // Initialize first two values in table
    catalan[0] = catalan[1] = 1;

    // Fill entries in catalan[] using recursive formula
    for (int i=2; i<=n; i++)
    {
        catalan[i] = 0;
        for (int j=0; j<i; j++)
            catalan[i] += catalan[j] * catalan[i-j-1];
    }

    // Return last entry
    return catalan[n];
}

// Driver program to test above function
int main()
{
    for (int i = 0; i < 10; i++)
        cout << catalanDP(i) << " ";
    return 0;
}
```

Output:

```
1 1 2 5 14 42 132 429 1430 4862
```

Time Complexity: Time complexity of above implementation is $O(n^2)$

Using Binomial Coefficient
We can also use the below formula to find nth catalan number in O(n) time.

[Tex] C_n = \frac{1}{n+1}{2n\choose n} [/Tex]

We have discussed a O(n) approach to find binomial coefficient nCr.

```
#include<iostream>
using namespace std;

// Returns value of Binomial Coefficient C(n, k)
unsigned long int binomialCoeff(unsigned int n, unsigned int k)
{
    unsigned long int res = 1;
```

```
    // Since C(n, k) = C(n, n-k)
    if (k > n - k)
        k = n - k;

    // Calculate value of [n*(n-1)*---*(n-k+1)] / [k*(k-1)*---*1]
    for (int i = 0; i < k; ++i)
    {
        res *= (n - i);
        res /= (i + 1);
    }

    return res;
}

// A Binomial coefficient based function to find nth catalan
// number in O(n) time
unsigned long int catalan(unsigned int n)
{
    // Calculate value of 2nCn
    unsigned long int c = binomialCoeff(2*n, n);

    // return 2nCn/(n+1)
    return c/(n+1);
}

// Driver program to test above functions
int main()
{
    for (int i = 0; i < 10; i++)
        cout << catalan(i) << " ";
    return 0;
}
```

Output:

```
1 1 2 5 14 42 132 429 1430 4862
```

Time Complexity: Time complexity of above implementation is O(n).

We can also use below formula to find nth catalan number in O(n) time.

[Tex]C_n = \frac{(2n)!}{(n+1)!\,n!} = \prod\limits_{k=2}^{n}\frac{n+k}{k} \qquad\mbox{ for }n\ge 0[/Tex].

References:
http://en.wikipedia.org/wiki/Catalan_number

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, MathematicalAlgo

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

Ted Wilson • a month ago

Here is the dynamic solution in Haskell:

```
catalan n = head $ catem n [1,1]
where catem n lis = if (length lis > n) then lis
else catem n ((sum $ zipWith (*) lis (reverse lis)):lis)
```


aa1992 • 7 months ago

excellent solution with o(n).

kaushik Lele • 9 months ago

@GeeksforGeeks
A full binary tree with 3 nodes can be formed in 6 ways.
But catalan number for n=3 is 5. Which does not match.
Is my understanding incorrect ?

For a binary tree with 3 nodes viz. (a,b,c) I could see below 6 combinations.

a ->b
-> c

a ->c

-> b

b ->a
-> c

b ->c
-> a

c ->a
-> b

c ->b
-> a

• Reply • Share ›

kaushik Lele kaushik Lele • 9 months ago

Article says that
"A rooted binary tree is full if every vertex has either two children or no children"

But the example given on http://www.findstat.org/Binary... page (which is related to Catalan number wikipedia)
gives examples of Binary tree with 3 nodes. But those are not full binary trees.

So
1) is requirement of full trees given in this article incorrect ?
2) We are just thinking on different arrangements of nodes where every node is treated same ?

• Reply • Share ›

Anurag Singh kaushik Lele • 8 months ago

It needs to be reworded.
On http://en.wikipedia.org/wiki/C...
"Successive applications of a binary operator can be represented in terms of a full binary tree.[3] (A rooted binary tree is full if every vertex has either two children or no children.) It follows that Cn is the number of full binary trees with n + 1 leaves:"
So here n = 3 is three operators. So 3 operators and 4 (i.e. n + 1) operands can be represented in 5 ways.

• Reply • Share ›

kaushik Lele Anurag Singh • 8 months ago

It still does not answer my question ->
What are those 5 ways of creating full binary tree?

The diagram there shows 5 Binary tree. But I see that there is only 1 tree which satisfies the condition of two child or no child which satisfies the condition of two-child-or-no-child.

Reply • Share ›

Anurag Singh → kaushik Lele • 8 months ago

Statement is: "It follows that Cn is the number of full binary trees with n + 1 leaves:"
i.e. total 2n + 1 nodes are involved while making the tree (n internal nodes as operator and n+1 leaf nodes as operands).

For n = 3 operator and n+1 = 4 operands
In the diagram, there are 3 operators (internal noes) and 4 operands (the leaves). Total 7 nodes are involved.

1 • Reply • Share ›

kaushik Lele → Anurag Singh • 8 months ago

Ohh .. its about "binary operator" !! I completely ignored that word and I was thinking in terms of normal Binary tree concept. Thanks for explaining patiently till I understood. **@GeeksforGeeks** please modify the sentence in this article to include word "operator" and add above explanation by **@Anurag Singh** This will help other learners.

Reply • Share ›

kaushik Lele → kaushik Lele • 8 months ago

Description in this article also mentions "n+1 leaves" but I by mistake took it as "n+1 total nodes" and caused all this confusion.

However it is better to explain it in terms of nodes than leaves; as depending on number on nodes; we can draw different arrangements. Leaves will automatically fall in place.

So we can say -> full binary tree with n-nodes (& hence n+1 leaves) can be formed with Cn ways :)

Reply • Share ›

Jun • 10 months ago

Method 2

http://ideone.com/YbNst4

Reply • Share ›

Jun • 10 months ago

Method 1

http://ideone.com/7q8wtV

Reply • Share ›

valar morghulis • 10 months ago

dp soln can be optimized:-
for (int j=0; j

3 ^ | v • Reply • Share ›

Manish M Berwani • a year ago

A better approach to this problem would be

C[n+1]= (2*(2*n + 1)*C[n])/(n+2)

3 ^ | v • Reply • Share ›

<HoldOnLife!#> ➦ Manish M Berwani • 10 months ago

It would also require O(n) only I guess?

^ | v • Reply • Share ›

arjomanD • a year ago

Good !
Does this site have contests ?

^ | v • Reply • Share ›

GOPI GOPINATH ➦ arjomanD • a year ago

No. Just Concepts

5 ^ | v • Reply • Share ›

Karshit Jaiswal ➦ GOPI GOPINATH • a year ago

which site are you talking about?

^ | v • Reply • Share ›

GOPI GOPINATH ➦ Karshit Jaiswal • a year ago

Geeksforgeeks.org

^ | v • Reply • Share ›

kabeer • a year ago

nice!

^ | v • Reply • Share ›

Vivek VV • a year ago

Please currect the spelling mistake in 2nd usage example :P

2 ^ | v • Reply • Share ›

GeeksforGeeks Mod ➦ Vivek VV • a year ago

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

^ | v • Reply • Share ›

Guest GeeksforGeeks • 6 months ago

Here T(n - i) should be changed to T(n - i - 1)

• Reply • Share ›

- 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

- ## Recent Comments

  - lt_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

-