Bell Numbers (Number of ways to Partition a Set)

Given a set of n elements, find number of ways of partitioning it.

Examples:

```
Input:  n = 2
Output: Number of ways = 2
Explanation: Let the set be {1, 2}
            { {1}, {2} }
            { {1, 2} }

Input:  n = 3
Output: Number of ways = 5
Explanation: Let the set be {1, 2, 3}
             { {1}, {2}, {3} }
             { {1}, {2, 3} }
             { {2}, {1, 3} }
             { {3}, {1, 2} }
             { {1, 2, 3} }.
```

Solution to above questions is Bell Number.

What is a Bell Number?

Let **S(n, k)** be total number of partitions of n elements into k sets. The value of n'th Bell Number is sum of S(n, k) for k = 1 to n.

$$\text{Bell}(n) = \sum_{k=0}^{n} S(n, k)$$

Value of S(n, k) can be defined recursively as, S(n+1, k) = k*S(n, k) + S(n, k-1)

How does above recursive formula work?

When we add a (n+1)'th element to k partitions, there are two possibilities.

1) It is added as a single element set to existing partitions, i.e, S(n, k-1)

2) It is added to all sets of every partition, i.e., k*S(n, k)

S(n, k) is called Stirling numbers of the second kind

First few Bell numbers are 1, 1, 2, 5, 15, 52, 203, ….

A **Simple Method** to compute n'th Bell Number is to one by one compute S(n, k) for k = 1 to n and return sum of all computed values. Refer this for computation of S(n, k).

A **Better Method** is to use Bell Triangle. Below is a sample Bell Triangle for first few Bell Numbers.

```
1
1 2
2 3 5
5 7 10 15
15 20 27 37 52
```

The triangle is constructed using below formula.

```
// If this is first column of current row 'i'
If j == 0
   // Then copy last entry of previous row
   // Note that i'th row has i entries
   Bell(i, j) = Bell(i-1, i-1)

// If this is not first column of current row
Else
   // Then this element is sum of previous element
   // in current row and the element just above the
   // previous element
   Bell(i, j) = Bell(i-1, j-1) + Bell(i, j-1)
```

Interpretation

Then Bell(n, k) counts the number of partitions of the set {1, 2, …, n + 1} in which the element k + 1 is the largest element that can be alone in its set.

For example, Bell(3, 2) is 3, it is count of number of partitions of {1, 2, 3, 4} in which 3 is the largest singleton element. There are three such partitions:

```
    {1}, {2, 4}, {3}
    {1, 4}, {2}, {3}
    {1, 2, 4}, {3}.
```

Below is Dynamic Programming based implementation of above recursive formula.

```
// A C++ program to find n'th Bell number
#include<iostream>
using namespace std;

int bellNumber(int n)
{
   int bell[n+1][n+1];
   bell[0][0] = 1;
   for (int i=1; i<=n; i++)
   {
      // Explicitly fill for j = 0
      bell[i][0] = bell[i-1][i-1];

      // Fill for remaining values of j
      for (int j=1; j<=i; j++)
         bell[i][j] = bell[i-1][j-1] + bell[i][j-1];
   }
```

```
    return bell[n][0];
}

// Driver program
int main()
{
    for (int n=0; n<=5; n++)
        cout << "Bell Number " << n << " is "
             << bellNumber(n) << endl;
    return 0;
}
```

Run on IDE

Output:

```
Bell Number 0 is 1
Bell Number 1 is 1
Bell Number 2 is 2
Bell Number 3 is 5
Bell Number 4 is 15
Bell Number 5 is 52
```

Time Complexity of above solution is $O(n^2)$. We will soon be discussing other more efficient methods of computing Bell Numbers.

Another problem that can be solved by Bell Numbers.
A number is squarefree if it is not divisible by a perfect square other than 1. For example, 6 is a square free number but 12 is not as it is divisible by 4.
Given a squarefree number x, find the number of different multiplicative partitions of x. The number of multiplicative partitions is Bell(n) where n is number of prime factors of x. For example x = 30, there are 3 prime factors of 2, 3 and 5. So the answer is Bell(3) which is 5. The 5 partitions are 1 x 30, 2 x15, 3 x 10, 5 x 6 and 2 x 3 x 5.

Exercise:
The above implementation causes arithmetic overflow for slightly larger values of n. Extend the above program so that results are computed under modulo 1000000007 to avoid overflows.

Reference:
https://en.wikipedia.org/wiki/Bell_number
https://en.wikipedia.org/wiki/Bell_triangle

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

Company Wise Coding Practice Topic Wise Coding Practice

7 Comments Category: Dynamic Programming Mathematical

Related Posts:

- Find all combinations of k-bit numbers with n bits set where 1 <= n <= k in sorted order
- Longest Geometric Progression
- Weighted Job Scheduling | Set 2 (Using LIS)
- Print Maximum Length Chain of Pairs
- Find Jobs involved in Weighted Job Scheduling
- Printing Longest Bitonic Subsequence
- Printing Maximum Sum Increasing Subsequence
- Find minimum adjustment cost of an array

(Login to Rate and Mark)

3 Average Difficulty : **3/5.0**
Based on **13** vote(s)

Add to TODO List
Mark as DONE

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

7 Comments **GeeksforGeeks** **Tejas Joshi**

Recommend **Share** **Sort by Newest**

Join the discussion…

icicle • 6 months ago
Hi How you get O(N^2) as the complexity of Bell number? Could you please explain?
• Reply • Share ›

NM • 8 months ago
For the nth bell number, we only need bell[n-1][n-1], so we can avoid the extra computation for bell[n][1..n] and return bell[n-1[n-1] directly. We'd need to specially handle n == 0 though.
• Reply • Share ›

Aditya Pal • 10 months ago
After reading this tutorial try NOVICE43 on Spoj
http://www.spoj.com/problems/N...
1 • Reply • Share ›

in time • 10 months ago
I have my solution for this prob call it as "intime number" :)

Imagine tree with root node with key as 1 and formula : node with key "k" will have (k + 1) children with first k childrens have key k and (k + 1)th child have key (k + 1)...now keep on expanding this tree till level N....and ans is number of leaf nodes at level N => By careful optimization we can reduce it to O(N) space and O(N^2) time complexity

• Reply • Share ›

in time ➚ in time • 10 months ago

Or Maybe we can even reduce it to O(N) time complexity using this logic...better than O(N^2) solution specified in article

• Reply • Share ›

kumar Shivam • 10 months ago

the first explanation for bell number 2 has set {2,2} instead of {1,2}. Kindly rectify it cas its creating confusion

• Reply • Share ›

GeeksforGeeks Mod ➚ kumar Shivam • 10 months ago

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

• Reply • Share ›