,9，排列与组合

2018年4月6日 15:29

$$C_{2n}^{n} - C_{2n}^{n+1} = \frac{1}{n+1} * C_{2n}^{n}$$

卡特兰数重要公式之一

$f(0)=1, f(1)=1, f(2)=2, f(3)=5$时

$$f(n)=f(0)*f(n-1)+f(1)*f(n-2)+f(3)*f(n-4)+ \cdots\cdots + \cdots\cdots + f(n-1)*f(0) = \frac{1}{n+1} * C_{2n}^{n}$$

卡特兰数重要公式2.

1，

屏幕剪辑的捕获时间: 2018/4/6 15:30,2，

```
/*
有n个信封，包含n封信，现在把信拿出来，再装回去，要求每封信不能装回它原来的信封，
问多少种装法？

解法：
这道题我们可以利用递归的解法，n封信假设有f(n)种装法，这里我们只考虑n>2的情况
首先，假设第n封信放入了第i个信封，那么有两种情况，
1，第i封信也放入了信封n，那么剩下的就是一个n-2的装信封问题，后续为f(n-2)
2.第i封信没放入信封n，那么就是一个n-1封信，对应的不放入各自信封问题。后续为f(n-1)
然后i的选择有n-1种（除了位置n），因此，总数为(n-1)*(f(n-1)+f(n-2))
f1 = 0,f2 = 1,
*/
#include<bits/stdc++.h>
using namespace std;
class Solution
{
```

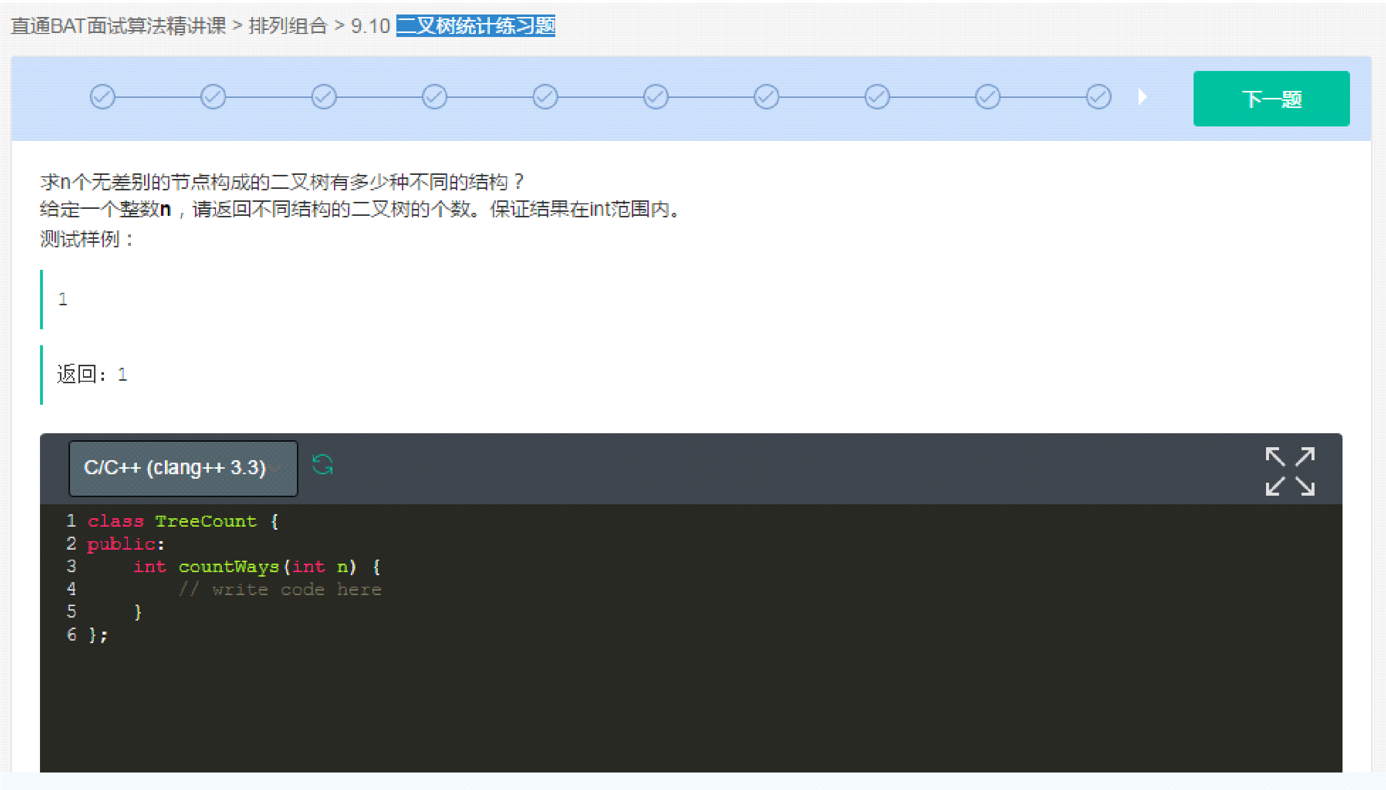
```
public:
    int solution(int n)
    {
        if(n == 1 || n == 2)
            return n-1;
        else
        {
            return (n-1)*(solution(n-1)+solution(n-2));
        }
    }
};


int main()
{
    int n = 4;//此时应该=9
    Solution s;
    cout<<s.solution(n)<<endl;
    return 0;

}
```

来自 <http://tool.oschina.net/highlight>

2，



屏幕剪辑的捕获时间: 2018/4/6 15:30

/*
问题：求n个无差别的节点构成的二叉树有多少种不同的结构

解法：
假设n个无差别的节点构成的不同结构数为f(n)
当n=0时，只有空树一种可能，f(0)=1,
n =1 时，也只有一种可能,f(1) = 1;
n = 2,f(2) = 2;
n = 3,f(3) = 5
现在我们把n个节点排成一排，
如果把第一个节点当做头，那么左子树为空（1种方法），右子树为剩下的n-1个节点。（f(n-1种方法），此时方法数为f(0)×f(n-1)
如果第二个节点为头，那么左子树为1,(1种方法），右子树为剩下的n-2个节点，则方法数为f(1)×f(n-2)
以此类推，第n个节点为头，那么左子树为f(n-1),右子树为f(0),方法数为f(n-1)*f(0)

加起来就是f(0)*f(n-1)+f(1)*f(n-2)+f(2)*f(n-3)+...+f(n-1)*f(0) = 1/(n+1) * c(2n,n)

以上，就是卡特兰数的一个性质，如果f0 = 1, f1 = 1,f2 = 2,f3 = 5,
且f(n) = f(0)*f(n-1)+f(1)*f(n-2)+f(2)*f(n-3)+...+f(n-1)*f(0)
那么就可以化简为1/(n+1) * c(2n,n)

因此，结构数为1/(n+1)*c(2n,n)
*/

来自 <http://tool.oschina.net/highlight>

3，

屏幕剪辑的捕获时间: 2018/4/6 15:31
/*
问题：在X*Y的方格中，以左上角格子为起点，右下角格子为终点，每次只能往下或往右走，请问
一共右多少种不同的走法？给定两个正整数int x, int y，返回走法数目，保证x+y <= 12

如：6*9的方格，每次都往下or往右走，一共有多少种走法？
首先，一共需要走5+8 = 13步，其中5步往下，8步往右(注意这里起点是在第一个格子里面)
答案是C(13,5) or C(13,8).

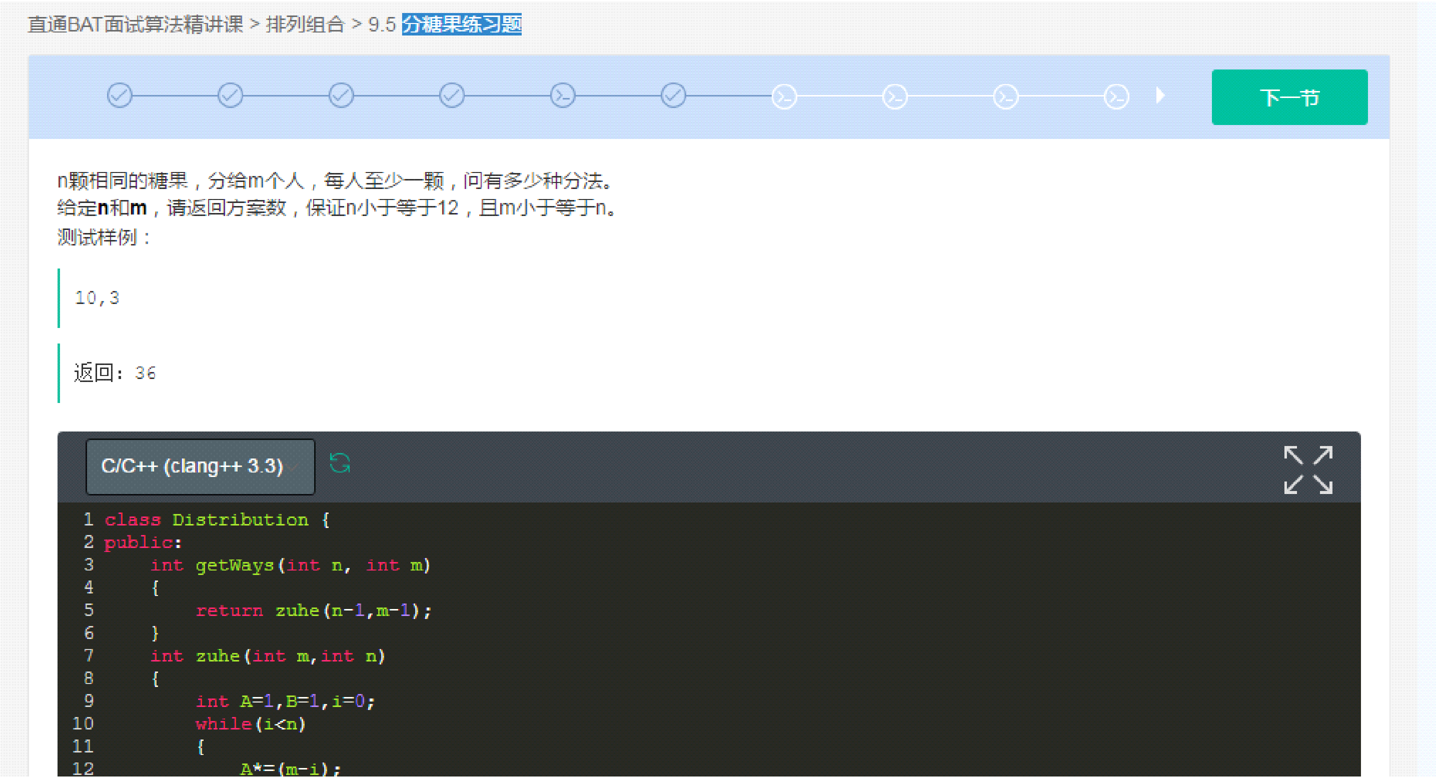
!!!此外，需注意的是，阶乘得到的结果很大，建议用longlong存

*/
```
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    long long calc(int x)//计算x的阶乘
    {
        if(x == 1)
            return 1;
        else
            return x*calc(x-1);
    }
    long long solution(int x, int y)
    {
        long long res;
        res = calc(x+y-2)/(calc(x-1)*calc(y-1));
        return res;
    }
};
```

```
int main()
{
    int x = 6;
    int y = 9;
    Solution s;
    cout<<s.solution(x,y)<<endl;
    return 0;
}
```

来自 <http://tool.oschina.net/highlight>

4，



屏幕剪辑的捕获时间: 2018/4/6 15:31

/*
问题：n颗相同的糖果，分给m个人，每人至少一颗，问有多少种分法？
给定n，m，返回方案数，保证n<=12(是为了计算阶乘不溢出），且m<=n
如，现在有10颗糖，3个人。

解法：10颗糖中间有9个中间位置，现在我们只需要从中选出两个放隔板，分成的三部分刚好是每人至少一颗的分法，
此时，方法数有C（9,2） = 36种

拓展：10个不同的球放入3个不同的桶里有多少种方法？

解法：每个球都有3种选择，因此有3^10次方种选择

拓展2，有10颗糖，如果每天至少吃一颗，吃完为止，问有多少种不同的吃法？

解法：如果想一天吃完，只有1种吃法
如果两天，那么相当于9个间隔中放一个隔板，有c(9,1)种吃法
三天：c(9,2)

n天：c(9,n-1),n<=10

加起来就是c(9,0)+c(9,1)+..+c(9,9) = 2^9

这里有个很重要的公式

c(n,0)+c(n,1)+...+c(n,n) = (1+1)^n = 2^n；二项式定理的内容。

```
*/
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    long long cald(int x)
    {
        if(x == 1)
            return 1;
        else
            return x*cald(x-1);
    }
    long long solution(int n, int m)
    {
        long long res;
        res = cald(n-1)/(cald(m-1)*cald(n-m));
        return res;
    }
};

int main()
{
    int n = 10;
    int m = 3;
    Solution s;
    cout<<s.solution(n,m)<<endl;
    return 0;
}
```

来自 <http://tool.oschina.net/highlight>

5，


直通BAT面试算法精讲课 > 排列组合 > 9.11 高矮排列练习题

下一题

12个高矮不同的人，排成两排，每排必须是从矮到高排列，而且第二排比对应的第一排的人高，问排列方式有多少种？
给定一个偶数**n**，请返回所求的排列方式个数。保证结果在int范围内。
测试样例：

1

返回：1

C/C++ (clang++ 3.3)

```
class HighAndShort
{
public:
    int countWays(int n)
    {
        return zuhe(n,n/2)/(n/2+1);
    }
    int zuhe(int m,int n)
        //n<=m
```

屏幕剪辑的捕获时间: 2018/4/6 15:31

/*
问题：12个高矮不同的人，排成两排，每排必须是从矮到高排，且第二排比对应的第一排的人高，求排列方式有多少种？
给定一个偶数n，返回所求的排列个数。

解法：这是一个隐藏很深的卡特兰数问题，我们将12个人从矮到高编号，
0表示第一排的人，1表示第二排的人。那么这个串中就包含6个0和6个1
如，0 0 0 0 1 1 0 1 0 1 1 1 表示前四个人在第一排，56在第二排，7在第一排，

8在第二排，9在第一排，10,11,12在第二排

如果有一个前缀1比0多，则说明第二排中的人多了，则后面必有1个位置，第二排没有1可以放了，是不合法的

因此，这个问题又变得跟左右括号合法一样的问题，任意前缀不能出现1比0多的情况
因此，排列方式为c(2n,n)- c(2n,n+1) = 1/ (n+1)*c(2n,n)

*/

来自 <http://tool.oschina.net/highlight>

6，

直通BAT面试算法精讲课 > 排列组合 > 9.4 孤傲的A练习题

下一题

A(A也是他的编号)是一个孤傲的人，在一个n个人(其中编号依次为1到n)的队列中，他于其中的标号为b和标号c的人都有矛盾，所以他不会和他们站在相邻的位置。现在问你满足A的要求的对列有多少种？
给定人数**n**和三个人的标号**A**,**b**和**c**，请返回所求答案，保证人数小于等于11且大于等于3。
测试样例：

```
6,1,2,3
```

```
288
```

C/C++ (clang++ 3.3)

```
class LonelyA {
public:
    int getWays(int n, int A, int b, int c)
    {
        int s1=pailie(n),s2=pailie(n-1),s3=pailie(n-2);
        return (s1-4*s2+2*s3);

    }
```

屏幕剪辑的捕获时间: 2018/4/6 15:31
/*
问题:
六个人排成一排，要求甲乙不相邻，甲丙也不相邻，求多少种排法?

首先六个人的全排有6!，
然后计算甲乙相邻的概率。甲乙or乙甲，2*5!
然后计算甲丙相邻的概率，也是2*5!
上面有重叠的地方，就是乙甲丙or丙甲乙相邻出现的情况，这种情况将三者看做一个整体，有2*4! 种
因此 6! -2*5! -2*5! +2*4! 种

A（也是他的编号）是一个孤傲的人，在一个n个人的队列中，它跟编号为b，c的人有矛盾，所以他不会yu其
站在相邻的位置，问满足A要求的队列有多少种
给定人数n和三个人的标号a，b，c，求返回所求答案
*/

来自 <http://tool.oschina.net/highlight>

7，

直通BAT面试算法精讲课 > 排列组合 > 9.8 进出栈练习题

下一题

n个数进出栈的顺序有多少种？假设栈的容量无限大。
给定一个整数**n**，请返回所求的进出栈顺序个数。保证结果在int范围内。
测试样例：

```
1
```

```
返回：1
```

C/C++ (clang++ 3.3)

```
class Stack {
public:
    int countWays(int n) {
        // write code here
    }
};
```

屏幕剪辑的捕获时间: 2018/4/6 15:32

/*
问题：n个数进出栈的顺序有多少种？假设栈的容量无限大
给定一个整数n，请返回所求的进出栈的顺序个数，保证结果在int范围内（这种情况通常要mod1000000007）

解法：首先，要出栈必须先进栈，因此这道题的解法和9_8很类似，我们把进栈记为1，出栈记为-1
那么方法数为c(2n,n)-c(2n,n+1) = 1/(n+1)*c(2n,n)
*/

来自 <http://tool.oschina.net/highlight>

8，

直通BAT面试算法精讲课 > 排列组合 > 9.7 括号序列练习题

下一题

假设有n对左右括号，请求出合法的排列有多少个？合法是指每一个括号都可以找到与之配对的括号，比如n=1时，()是合法的，但是)(为不合法。
给定一个整数**n**，请返回所求的合法排列数。保证结果在int范围内。
测试样例：

```
1
```

```
返回：1
```

C/C++ (clang++ 3.3)

```
class Parenthesis {
public:
    int countLegalWays(int n) {
        // write code here
    }
};
```

屏幕剪辑的捕获时间: 2018/4/6 15:32

/*问题：假设有n对括号，求出合法的排列有多少个？合法是指每个括号都能找到与之配对的括号，比如n=1时，（）是合法的，)(是不合法的
给定一个整数n，求返回所求的合法排列数。保证结果在int范围内。

解法：
首先，左括号数为n，右括号数为n，总的排列数为c（2n，n）。
而不合法的排列，一定是出现了右括号比左括号多一个 的前缀。如())... 或）...
这里我们将(记为1，)记为-1，
如果括号对数为3，
那么())(() 为1 -1 -1 1 1 -1，明显前三个这个前缀是不合法的。
我们将不合法的部分符号颠倒，变成-1 1 1 1 1 -1 ，如此得到n-1个-1和n+1个1组成的排列
可以证明，每个非法的排列通过如上的变换，都可以得到唯一一个n+1个左括号和n-1个右括号组成的排列。证明略
因此，我们可以这样说，有且只有n-1个右括号和n+1个左括号组成的排列是非法的。

因此，合法的对数等于c(2n,n)- c(2n,n+1) = 1/(n+1)*c(2n,n) 这个公式是由卡特兰数重要公式得到的。
*/

```
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    int cald(int x)//计算x的阶乘
    {
        if(x == 1)
            return 1;
        else
            return x*cald(x-1);
    }

    int solution(int n)
    {
        int res = cald(2*n)/(cald(n)*cald(n));//尽量避免长计算。
        return res/(n+1);
    }
};

int main()
{
    int n = 1;
    Solution s;
    cout<<s.solution(n)<<endl;
    return 0;
}
```

来自 <http://tool.oschina.net/highlight>

,9 ,

直通BAT面试算法精讲课 > 排列组合 > 9.9 排队买票练习题

下一题

2n个人排队买票，n个人拿5块钱，n个人拿10块钱，票价是5块钱1张，每个人买一张票，售票员手里没有零钱，问有多少种排队方法让售票员可以顺利卖票。
给定一个整数**n**，请返回所求的排队方案个数。保证结果在int范围内。
测试样例：

```
1
```

返回：1

C/C++ (clang++ 3.3)

```cpp
class BuyTickets {
public:
    int countWays(int n) {
        // write code here
    }
};
```

屏幕剪辑的捕获时间: 2018/4/6 15:32

/*
问题：2n个人买票，n个人拿5块，n个人拿10块，票价是5块一张，每人买一张票，售票员手里没有零钱，问有多少种方法能让售票员顺利卖票
给定要给整数n，返回所求的方案个数

解法：这道题仍然跟括号合法的排列数类似，拿5块的记为1，拿10块的记为-1.
方法数为c(2n, n) - c(2n, n+1) = 1/n+1 * c(2n, n)
*/

来自 <http://tool.oschina.net/highlight>

10，

直通BAT面试算法精讲课 > 排列组合 > 9.3 站队问题练习题

下一题

n个人站队，他们的编号依次从1到n，要求编号为a的人必须在编号为b的人的左边，但不要求一定相邻，请问共有多少种排法？第二问如果要求a必须在b的左边，并且一定要相邻，请问一共有多少种排法？
给定人数**n**及两个人的编号**a**和**b**，请返回一个两个元素的数组，其中两个元素依次为两个问题的答案。保证人数小于等于10。
测试样例：

```
7,1,2
```

返回：[2520,720]

C/C++ (clang++ 3.3)

```cpp
class StandInLine {
public:
    vector<int> getWays(int n, int a, int b) {
        // write code here
    }
};
```

屏幕剪辑的捕获时间: 2018/4/6 15:32

```
/*
问题：n个人站队，他们编号从1到n，要求编号为a的人必须在编号为b的人左边，但不要求相邻，请问一共有多少种排法？
第二问：如果要求a必须在b的左边，且相邻，请问一共有多少种排法？

给定人数n和两个人的编号a，b，请返回一个两个元素的数组，其中两个元素依次为两个问题的答案，保证人数小于等于10.


首先问题1，n个人排队，一定有n！种站法，其中一半情况是A在B左边，另一半情况是B在A左边，因此，有n！/2种排法
第二问也很简单，A在B左边，可以将两者看做一个人，那么就是n-1个人的全排列。为（n-1）！



*/
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    long long cald(int x)//计算x的阶乘,虽说n<=10,但以防万一，还是用longlong存
    {
        if(x == 1)
            return 1;
        else
            return x*cald(x-1);
    }
    vector<long long> solution(int n,int a,int b)
    {
        long long res1 = cald(n)/2;
        long long res2 = cald(n-1);
        vector<long long> res;
        res.push_back(res1);
        res.push_back(res2);
        return res;
    }
};
int main()
{
    int n = 7;
    int a = 1;
    int b = 2;
    Solution s;
    vector<long long> res = s.solution(n,a,b);
    vector<long long>::iterator iter;
    for(iter = res.begin();iter!= res.end();iter++)
        cout<<*iter<<" ";
    cout<<endl;
    return 0;
}
```

来自 <http://tool.oschina.net/highlight>