算法基础（二十七）：数学基础 - 组合数 - 卡特兰数

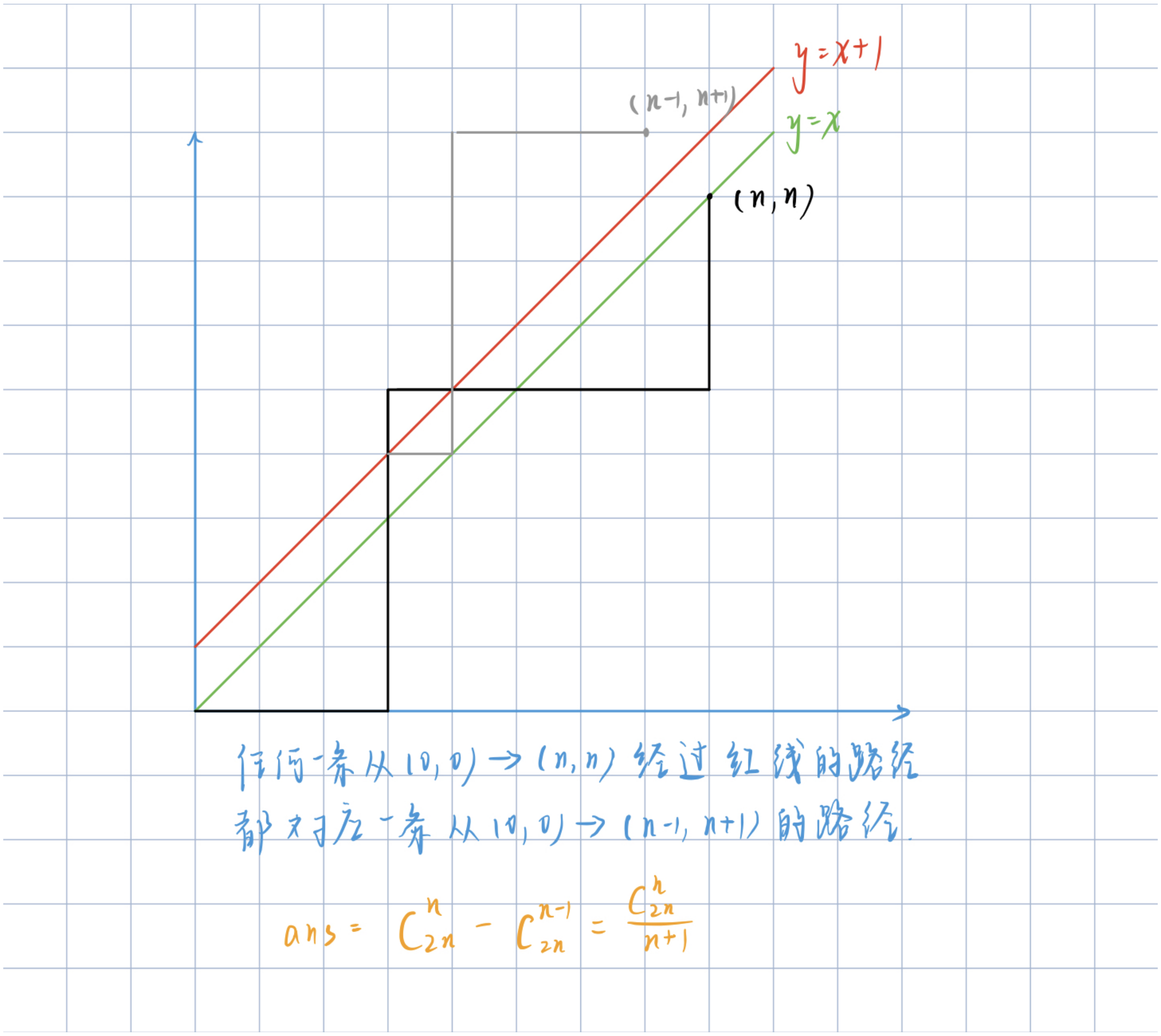
卡特兰数就是：

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

很多方案都是卡特兰数的结果，这里证明略，下面看一个使用的例子

> 给定 n 个 0 和 n 个 1，它们将按照某种顺序排成长度为 $2n$ 的序列，求它们能排列成的所有序列中，能够满足任意前缀序列中 0 的个数都不少于 1 的个数的序列有多少个。
>
> 输出的答案对 10^9+7 取模。
>
> **输入格式**
>
> 共一行，包含整数 n。
>
> **输出格式**
>
> 共一行，包含一个整数，表示答案。
>
> **数据范围**
>
> $1\le n\le 10^5$
>
> **输入样例：**
>
> ```
> 3
> ```
>
> **输出样例：**
>
> ```
> 5
> ```

我们将`0 1`置于坐标系中，`0`表示向右走，`1`表示向上走，那么当`n = 6`的时候，排成长度为`12`的序列，其中有`6`个`1`，`6`个`0`，等价到坐标系中就是走到坐标为`(6, 6)`这个位置



图片来自Acwing用户：番茄酱

要保证仍和序列中`0`的个数不少于`1`的个数，就等价于走到`(6，6)`这个点的路线必须在上图的绿色的线下面（包括绿色的线），也就是得走到图中红色的线下面。其中走到`(6，6)`的总的路线数是C_{12}^{6}（在12步中挑出6步向上走），然后用总的路线减去经过红色不合法的路线即可

如果一条路线经过了红色到达点`(6，6)`，做对称可以发现他经过走到了点`(5，7)`，而如果一个路线走到了`(5，7)`，作对称可以发现其一定是经过红色走到了`(6，6)`，那就说明经过红色走到了`(6，6)`与走到点`(5，7)`是等价的，所以不经过红色走到了`(6，6)`的路线数就是$C_{12}^{6} - C_{12}^{5}$，用`n`表示就是$C_{2n}^{n} - C_{2n}^{n-1}$，也就是卡特兰数。

代码实现：

```
#include<iostream>
#include<algorithm>

using namespace std;

const int mod = 1e9 + 7;

typedef long long LL;

```

```
//快速幂求逆元
int qmi(int a, int k, int p)
{
    int res = 1;
    while(k)
    {
        if(k & 1) res = (LL) res * a % p;
        a = (LL) a * a % p;
        k >>= 1;
    }

    return res;

}

int main()
{
    int n;
    cin >> n;
    int a = 2 * n, b = n;
    int res = 1;
    //计算a * (a - 1) * ...(a - b + 1)
    for(int i = a; i > a - b; i --) res = (LL) res * i % mod;
    //计算除以b!，当然在此处是乘以他的逆元
    for(int i = 1; i <= b; i ++) res = (LL) res * qmi(i, mod - 2, mod) % mod;

    //在最后还要除以(n + 1)，也就是乘以(n + 1) 的逆元
    res = (LL) res * qmi(n + 1, mod - 2, mod) % mod;

    cout << res << endl;
    return 0;

}
```