

算法基础（二十三）数学基础 - 组合数 - 递推求法

组合数的基本公式为： $C_a^b = \frac{a*(a-1)*...(a-b+1)}{1*2*3*...b} = \frac{a!}{b!(a-b)!}$

当数据的范围过大的时候，比如以下的数据范围：

数据范围

$$1 \leq n \leq 10000,$$
$$1 \leq b \leq a \leq 2000$$

表示有 n 次查询，每次查询数据的最大范围是 2000，在每次查询中最坏的计算次数是 2000 次（2000!），那么总的计算次数就是2亿次，显然次数极大，我们需要寻找别的方式来求解组合数

所以这里就是根据不同的数据范围来介绍不同的求解组合数的方法

以下面这个题目为例：

输入格式

第一行包含整数 n 。

接下来 n 行，每行包含一组 a 和 b 。

输出格式

共 n 行，每行输出一个询问的解。

数据范围

$$1 \leq n \leq 10000,$$
$$1 \leq b \leq a \leq 2000$$

输入样例：

```
3
3 1
5 3
2 2
```

输出样例：

```
3
10
1
```

我们通过公式

$$C_a^b = C_{a-1}^b + C_{a-1}^{b-1}$$

可以将所有的组合数递推，关于这个公式的证明，将左右两边直接展开即可，也可以这样理解：

假设 a 个苹果中有一个苹果 e ，那么从 a 个苹果中选出 b 个苹果就可以分为包括 e 和不包括 e 两种方案，对于第一种方案的种类就是在剩下的 $a - 1$ 个苹果中选 b 个苹果，也就是 C_{a-1}^b ，第二种方案就是在剩下的 $a - 1$ 个苹果中选 $b - 1$ 个苹果，也就是 C_{a-1}^{b-1} ，从而证明了公式。

其实利用这种递推的思想就是先预处理一些东西，这样可以减少算法的复杂度

代码实现：

```
1  #include<iostream>
2  #include<algorithm>
3  using namespace std;
4
5  const int N = 2010, mod = 1e9 + 7;
6
7  long long c[N][N];
8
9  void init()
10 {
11     for(int i = 0; i < N; i++)
12         for(int j = 0; j <= i; j++)
13             if(!j) c[i][j] = 1; //如果j为0，就说明从i个苹果中选0个的方案为1
14             else c[i][j] = (c[i - 1][j] + c[i - 1][j - 1]) % mod; //答案取模
15 }
16
17 int main()
18 {
19     //初始化预处理
20     init();
21
22     int n;
23     scanf("%d", &n);
24     while(n --)
25     {
26         int a, b;
27         scanf("%d%d", &a, &b);
28         printf("%d\n", c[a][b]);
29     }
30
31     return 0;
32 }
```

注意这个方法有点动态规划的思想，时间复杂度是 $O(N^2)$ 的级别