[第一类Stirling数和第二类Stirling](http://blog.csdn.net/acdreamers/article/details/8521134)

**第一类Stirling数 s(p,k)**

**s(p,k)的一个的组合学解释是：将p个物体排成k个非空循环排列的方法数。**

**s(p,k)的递推公式： s(p,k)=(p-1)\*s(p-1,k)+s(p-1,k-1) ,1<=k<=p-1**

**边界条件：s(p,0)=0 ,p>=1  s(p,p)=1  ,p>=0**

**递推关系的说明：**

**考虑第p个物品，p可以单独构成一个非空循环排列，这样前p-1种物品构成k-1个非空循环排列，方法数为s(p-1,k-1)；**

**也可以前p-1种物品构成k个非空循环排列，而第p个物品插入第i个物品的左边，这有(p-1)\*s(p-1,k)种方法。**

**第二类Stirling数 S(p,k)**

**S(p,k)的一个组合学解释是：将p个物体划分成k个非空的不可辨别的（可以理解为盒子没有编号）集合的方法数。**

**k!S(p,k)是把p个人分进k间有差别(如：被标有房号）的房间(无空房）的方法数。**

**S(p,k)的递推公式是：S(p,k)=k\*S(p-1,k)+S(p-1,k-1) ,1<= k<=p-1**

**边界条件：S(p,p)=1 ,p>=0    S(p,0)=0 ,p>=1**

**递推关系的说明：**

**考虑第p个物品，p可以单独构成一个非空集合，此时前p-1个物品构成k-1个非空的不可辨别的集合，方法数为S(p-1,k-1)；**

**也可以前p-1种物品构成k个非空的不可辨别的集合，第p个物品放入任意一个中，这样有k\*S(p-1,k)种方法。**

**第一类斯特林数和第二类斯特林数有相同的初始条件，但递推关系不同。**

**题意：给N个元素，让我们求K个环排列的方法数。**

**斯特林第一类数的第推公式：**

**S（N，0）=0; S（N，N）=1； S（0，0）=0； S（N，K）=S（N-1，K-1）+S（N-1，K）\*（N-1）；**

**这个公式的意思是：当前N-1个数构成K-1 个环的时候，加入第N个 ，N只能构成单环！—S（N-1，K-1）如果N-1个数构成K个环的时候，加入第N个，N可以任意加入，N-1内的一个环里，所以（N-1）\*S（N-1，K）这个题目里，因为不能破坏第1个门：所以 S（N，K）-S（N-1，K-1）才是能算构成K个环的方法数！就是去掉1自己成环的情况 。**

1. #define N 21
2. **\_\_int64** fac[N]={1,1};
3. **\_\_int64** stir[N][N];
5. **void** init()  {
6. **int** i, j;
7. **for**(i=2;i<N;i++)  fac[i]=i\*fac[i-1];
8. memset(stir,0,**sizeof**(stir));
9. stir[0][0]=0;
10. stir[1][1]=1;
11. **for**(i=2;i<N;i++)
12. **for**(j=1;j<=i;j++)
13. stir[i][j]=stir[i-1][j-1]+(i-1)\*stir[i-1][j];
14. }
15. **int** main()  {
16. init();**int** t;
17. scanf("%d",&t);
18. **while**(t--)  {
19. **int** n, k, i;
20. scanf("%d %d",&n,&k);
21. **\_\_int64** cnt=0;
22. **for**(i=1;i<=k;i++)
23. cnt+= stir[n][i] - stir[n-1][i-1];//注意：去掉1自己成环的
24. printf("%.4lf\n",1.0\*cnt/fac[n]);
25. }
26. **return** 0;

}