**冒泡排序计数**

考虑冒泡排序的一种实现。  
　　bubble-sort (A[], n)  
　　> round = 0  
　　> while A is not sorted  
　　> > round := round + 1  
　　> > for i := 1 to n - 1  
　　> > > if (A[i] > A[i + 1])  
　　> > > > swap(A[i], A[i + 1])  
　　求1 .. n的排列中，有多少个排列使得A被扫描了K遍，亦即算法结束时round == K。  
  
　　答案模20100713输出。

输入格式

　　输入包含多组数据。每组数据为一行两个整数N，K。

输出格式

　　对每组数据，输出一行一个整数表示答案。

样例输入

3  
3 0  
3 1  
3 2

样例输出

1  
3  
2

数据规模和约定

　　T <= 10 ^ 5。  
　　1 <= K < N < 10 ^ 6。

锦囊1

挖掘试题的性质，使用组合公式求解。

锦囊2

注意到一个性质：对排列P，设C[i]表示P[1 .. i - 1]中大于P[i]的数的个数，则我们只需要统计使得max(C[i]) = K的P的个数。因为可以注意到在每次扫描中，新的C'[i] = max(0, C[i] - 1)，而C[1 .. n]均为零是排列有序的充要条件。 考虑这一统计。我们尝试统计max(C[i]) <= K的排列个数ans(K)，之后ans(K) - ans(K - 1)即为所求。 考虑max(C[i]) <= K的排列的构造方法。从空序列开始，我们以任意顺序插入最大的K个数，于是这一部分的方案数为K!。之后，首先插入第K + 1大的数，它可以插入在任意位置；插入第K + 2大的数时，由于它的位置的C值不能超过K，而现有的数都比它大，它只能插入在前K个数之间的K + 1个位置之一；第K + 3大至第N大的数的插入与此同理。因此，插入后N - K个数的方案为(K + 1) ^ (N - K)。所以ans(K) = K! \* (K + 1) ^ (N - K)，原题所求为K! \* ((K + 1) ^ (N - K) - K ^ (N - K))。

C++代码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define LF\_PRECISION 10

#define INT\_64\_MOD "%I64d"

#define UNSIGNED\_64\_MOD "%I64u"

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<complex>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define FAST\_RW ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(ll i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(ll i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(ll i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(ll i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(ll i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define ll long long

#define ull unsigned long long

#define lf long double

#define pc putchar

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define pii pair<int,int>

#define pdd pair<double,double>

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define ms(x,v) memset((x),(v),sizeof(x))

#define mc(x,y) memcpy((x),(y),sizeof(y))

#define NL puts("");

#define fin(x,c) ((c).find(x)!=(c).end())

using namespace std;

template<class T1,class T2,class T3>

bool \_IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

ull gcd(ull a,ull b){

if(!b)return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#define NOT\_TESTING\_TEMPLATE\_CPP

#include"wmx16835.cpp"

#else

int ebtpqJsBCnTgggi;

#define LOG {

#define TEL }

#define SHOW\_TIME

#define test(...) ebtpqJsBCnTgggi

#define TEST(...) ebtpqJsBCnTgggi

#define TRY(...)

#define PF

#define PP ;

#endif

bool S(char\*a){

return scanf("%s",a)==1;

}

char DATaJNTFnlmAoya[2];

template<class T>

bool S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c")){

if(scanf("%1s",DATaJNTFnlmAoya)==-1)

return 0;

a=\*DATaJNTFnlmAoya;

return 1;

}

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else if(!strcmp(x,"e"))return (cin>>a)!=0;

else test("Input format error!\n");

}

void \_P(string x){

printf("%s",x.c\_str());

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else if(!strcmp(x,"e"))cout<<setprecision(LF\_PRECISION)<<a;

else test("Output format error!\n");

}

template<class T1,class T2>

bool S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

bool S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

bool S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

bool S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

pc(' ');

}

template<class T1,class T2>

void P(T1 a,T2 b){

\_P(a);pc(' ');

\_P(b);pc(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);pc(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);pc(' ');

\_P(b);pc(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);pc(' ');

\_P(b);pc(' ');

\_P(c);pc(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);pc(' ');

\_P(b);pc(' ');

\_P(c);pc(' ');

\_P(d);pc(' ');

\_P(e);NL

}

template<class T>

void PA(T\*a,int n,char c=' '){

FR(i,n-1)\_P(a[i]),pc(c);

PN(a[n-1]);

}

template<class T>

void PA(const T&x,char c=' '){

IT(x) ita=x.begin();

FE(it,x){

\_P(\*it);

if(++ita==x.end())NL

else pc(c);

}

}

int kase;

const double pi=4\*atan(1);

const double ep=1e-9;

//}

ll mod=20100713;

ll jc[1000005];

void init(){

jc[0]=1;

FOR(i,1000000)

jc[i]=i\*jc[i-1]%mod;

}

ll ksm(ll x,int t){

ll res=1,tmp=x;

while(t){

if(t&1)res=res\*tmp%mod;

tmp=tmp\*tmp%mod;

t>>=1;

}

return res;

}

int main(){

SHOW\_TIME

init();

int t;

S(t);

while(t--){

ll n,k;

S(n,k);

ll res=jc[k]\*ksm(k+1,n-k)-jc[k-1]\*ksm(k,n-k+1);

PN((res%mod+mod)%mod);

}

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.43 build 20150130\*\*\*\*\*\*\*\*\*\*\*\*\*/

C代码

Java代码