**再帰を使った組み合わせ計算の関数**

long combination(long n,long r){

if(r==0 || n==r)return 1;

return combination(n-1,r)+combination(n-1,r-1);

}

ここでは を用いている。この等式は次の式変形で証明できる。

また、 なのでこのどちらかになったら1を返す。それまではひたすら分解し続けるというプログラム。数が多いほど展開する回数も多くなるので大きい数の計算はできない。

**再帰を使わないで計算する関数**

long combination(long n,long r){

if(r==0 || n==r)return 1;

int i=0;

double c=1;

if(n-r<r)r=n-r;

for(i=0;i<r;i++)c=c\*(double)(n-i)/(double)(r-i);

return (long)c;

}

ここでは一度に階乗の計算を行うと大きい数の場合変数が所持できる数をはるかに超えるので次のような計算をやっている。

ここでを割り切れない数がきたときのためにdouble型で計算を行い最後にlong型にキャストして返している。さらに次の等式

を用いて計算数を減らすようにしている。