这两天编写的两个程序：

1. Perm.java
2. LoadingCorrectEdition.java

让我对生成排列树或者n叉子集树有了新的认识。

下面是这两个程序的源代码：

# Perm.java

public static void perm(int [] a,int start,int end){

if(start==end){

i++;

System.out.println("第"+i+"个排列为：");

for(int i=0;i<a.length;i++)

System.out.println(a[i]);

return;

}

for(int i=start;i<=end;i++){

swap(a,i,start);

perm(a,start+1,end);

//swap(a[i],a[start]);

swap(a,i,start);

}

}

public static void swap(int [] a,int i,int j){

int temp;

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

# LoadingCorrectEdition.java

public void BackTrack(int i){

if(i>=n){

System.out.println("-------------------");

System.out.println("第"+count+++"个结果");

System.out.println("cw当前值"+cw);

for(int j=0;j<solutions.length;j++)

System.out.println(solutions[j]);

if(cw>bestW){

bestW=cw;

}

System.out.println("bestW当前值"+bestW);

System.out.println("-------------------");

return;

}

for(int k=0;k<=1;k++){

solutions[i]=k;

if(cw+k\*w[i]<=c){//注意这个写法的精妙所在：k=0,则向右子树遍历。k=1则向左子树遍历

cw+=k\*w[i];

BackTrack(i+1);

cw-=k\*w[i];

}

}

}

# 实现总结

无论是生成排列树，还是子集树，都不可避免地需要在回溯函数中加入for循环，

n叉子集树的for循环，循环的次数即为n,它的结点的个数为2^n

排列树的for循环的次数是递减的，它的节点的 个数为n!