

# JAVA Prufer序列

## 题意

给出标号从1到n的点，以及某些点最终的度数。

问有多少种度数满足要求的树

## 思路

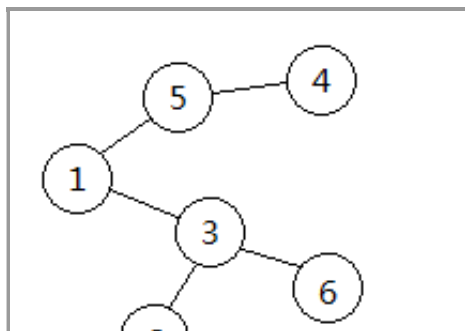
学习一种无根树的表达方法：**prufer序列**

### 将树转化成Prufer数列的方法

一种生成Prufer序列的方法是迭代删点，直到原图仅剩两个点。

对于一棵顶点已经经过编号的树T，顶点的编号为 $\{1, 2, \dots, n\}$ ，在第i步时，移去所有叶子节点（度为1的顶点）中标号最小的顶点和相连的边，并把与它相邻的点的编号加入Prufer序列中，重复以上步骤直到原图仅剩2个顶点。

#### 例子



首先在所有叶子节点中编号最小的点是2，和它相邻的点的编号是3，将3加入序列并删除编号为2的点。

接下来删除的点是4，5被加入序列，然后删除5，1被加入序列，1被删除，3被加入序列。

此时原图仅剩两个点（即3和6），Prufer序列构建完成，为 $\{3, 5, 1, 3\}$

### 将Prufer数列转化成树的方法

1. 设 $\{a_1, a_2, \dots, a_{n-2}\}$ 为一棵有n个节点的树的Prufer序列
2. 另建一个集合G含有元素 $\{1..n\}$
3. 找出集合中最小的未在Prufer序列中出现过的数，将该点与Prufer序列中首项连一条边，并将该点和Prufer序列首项删除，重复操作 $n-2$ 次
4. 将集合中剩余的两个点之间连边即可。

#### 例子

仍为上面的树，Prufer序列为 $\{3, 5, 1, 3\}$ ，开始时 $G = \{1, 2, 3, 4, 5, 6\}$ ，未出现的编号最小的点是2，将2和3连边，并删去Prufer序列首项和G中的2。接下来连的边为 $\{4, 5\}, \{1, 5\}, \{1, 3\}$ ，此时集合G中仅剩3和6，在3和6

之间连边，原树恢复。

该题需要将树转化成prufer编码：

$n$ 为树的节点数， $d[i]$ 为各节点的度数， $m$ 为无限制度数的节点数， $tot$ 为度数确定的点占prufer序列的长度。

$$tot = \sum_{i=1}^n d[i] - 1$$

则

所以要求在 $n-2$ 大小的数组中插入 $tot$ 个序号，共有  $C_{n-2}^{tot}$  种插法；

在 $tot$ 各序号排列中，插第一个节点的方法有  $C_{tot}^{d1-1}$  种插法；

插第二个节点的方法有  $C_{tot-(d1-1)}^{d2-1}$  种插法；

.....

另外还有 $m$ 各节点无度数限制，所以它们可任意排列在剩余的 $n-2-tot$ 的空间中，排列方法总数为

$$m^{n-2-tot};$$

根据乘法原理：

$$ans = C_{n-2}^{tot} * C_{tot}^{d1-1} * C_{tot-(d1-1)}^{d2-1} * \dots * C_{dn-1}^{dn-1} * m^{n-2-tot}$$

$$\Leftrightarrow \frac{(n-2)!}{(n-2-tot)!} * \frac{tot!}{(d1-1)!(tot-d1+1)!} * \dots * \frac{(dn-1)!}{(dn-1)!0!} * m^{n-2-tot}$$

$$\Leftrightarrow \frac{(n-2)! * m^{n-2-tot}}{(n-2-tot)! * (d1-1)! * (d2-1)! * \dots * (dn-1)!}$$

## Java知识点

1. Scanner cin = new Scanner(System.in);
2. 输入n: n = cin.nextInt();
3. 定义数组：int[] list2 = new int [2005];
4. 大数操作
  - (1) int 转BigInteger：BigInteger.valueOf();
  - (2) 大数a的运算：× a.multiply()  
÷ a.divide()
  - (3) 大数 a 转string（为了输出） a.toString();

5. 输出: System.out.print();

提交的时候类名要为Main , 写代码的时候不能为Main

## 代码

```
1  import java.util.Scanner;
2  import java.math.*;
3  import java.text.*;
4  public class Ccc
5  {
6      public static void main(String[] args)
7      {
8          Scanner cin = new Scanner(System.in);
9          int n,a;
10         int m,tot,num;
11         int[] list2 = new int [2005];
12         num = 0;
13         m = 0;
14         tot = 0;
15         n = cin.nextInt();
16         for (int i = 1;i<= n ;i++ )
17         {
18             a = cin.nextInt();
19             if (a == -1)
20             {
21                 m++;
22             }
23             else
24             {
25                 list2[++num] = a-1;
26                 tot = tot + a - 1;
27             }
28         }
29
30         BigInteger ans = new BigInteger("1");
31
32         for (int i = 2;i<=n-2 ;i++ )
33         {
34             BigInteger ii = BigInteger.valueOf(i);
35             ans = ans.multiply(ii);
```

```
36     }
37     int mm = n - 2 - tot;
38     for(int i = 1;i<=mm;i++)
39     {
40         ans = ans.multiply(BigInteger.valueOf(m));
41     }
42     for (int i = 2;i<=mm ;i++ )
43     {
44         BigInteger ii = BigInteger.valueOf(i);
45         ans = ans.divide(ii);
46     }
47
48     for(int i = 1;i<=num;i++)
49     {
50         for(int j = 2;j<=list2[i];j++)
51         {
52             BigInteger jj = BigInteger.valueOf(j);
53             ans = ans.divide(jj);
54         }
55     }
56     System.out.print(ans.toString());
57 }
58 }
```