# 1 题目简介

输入一组通信符号的使用频率

例：2 3 5 7 11 13 17 19 23 29 31 37 41

求各通信符号对应的前缀码。

# 2 解题思路及算法分析

上述问题是最优二元树在通信编码问题中生成最佳前缀码的典型应用，其核心思想是：用较多的数字编码频率权值低的节点，而用较少的数字编码频率较高的节点，这样一来，最终编码需要的数字总长度等于节点编码数字长度×节点频率，长度大的编码能够乘小频率，频率大的节点能够乘小长度，从而使总体编码长度最短。

接下来用程序来实现上述思路：

1. 首先对输入的频率按照从小到大的顺序排序，这里采用了快速排序算法：

快速排序的核心思想是：通过一趟排序将待排记录分隔成独立的两部分，其中一部分记录的关键字均比另一部分的关键字小，则可分别对这两部分记录继续进行排序，以达到整个序列有序。

快速排序的具体流程是：(递归思路)

首先设置两个变量 left、right，排序开始时：left=0，right=num-1。接着确定整个数组的基准位置，所有元素比基准值小的摆放在基准前面，所有元素比基准值大的摆在基准的后面。本程序中默认数组的第一个数为基准数据，赋值给pivot，即pivot=array[left]。因为默认数组的第一个数为基准，所以接下来从left+1开始向后搜索，直到到达右边界，把比基准数小的数字调到左边位置，相应的，较大的数字调到右边位置，而且最后把基准数移动到这两个序列的中间位置，并计数交换次数。接下来递归调用快速排序函数，排序分界点前和分界点后的子数组排序，最终就会得到排序好的数组。

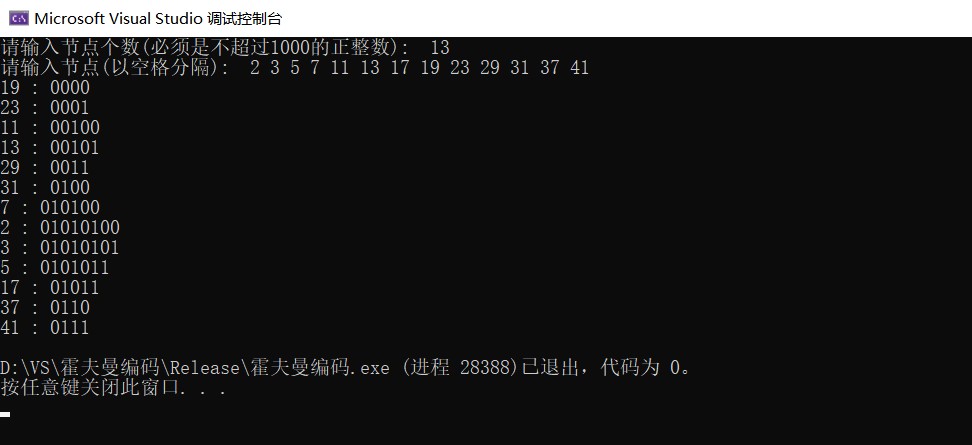
1. 接着生成最优二叉树：

在所有入度为0的顶点(不一定是树叶)中选出两个权最小的顶点, 添加一个新分支点, 以这2个顶点为儿子, 其权等于这2个儿子的权之和。接着利用循环结构重复此步骤，直到只有1个入度为0 的顶点为止。

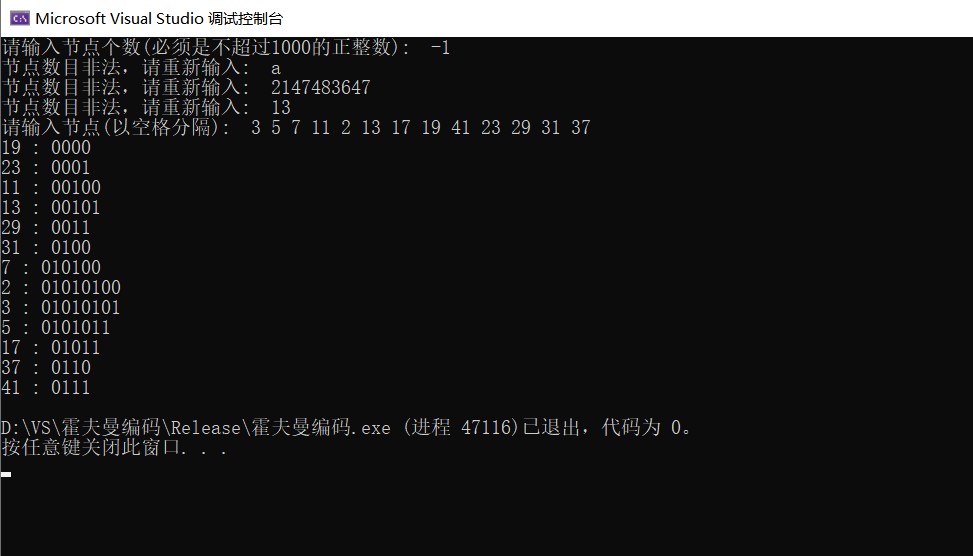
1. 最后通过遍历树，生成并输出各节点的编码：

本程序中使用了前序遍历（遍历的方式不影响编码结果）。对每个分支点, 若关联2条边, 则给左边标0, 右边标1; 若只关联1条边, 则可以给它标0(看作左边), 也可以标1(看作右边).。将从树根到每一片树叶的通路上标的数字组成的字符串便构成一个前缀码。

# 实验结果：



此外，为了提升程序的实用性与健壮性，正如上文对快速排序算法的介绍所述，本程序一方面在原有代码的基础上通过快速排序将节点频率按照从小到大的顺序排列，这样一来，用户在输入数字序列之前就不需要人为再进行排序，排序操作可以通过程序来实现；另一方面，本程序在输入节点个数这一步添加了对非法输入字符的处理功能，若是用户意外输入了非数字字符或者输入的数字超过限定的范围，则需要重新输入（综合考虑实际情况以及程序执行时间，本程序将节点数量限定在[1,1000]）,下面是测试结果：



需要注意的是：当新添加一个新分支点之后，该新分支点会作为中间运算结果成为一个单独的新节点代表其子节点参与编码，所以，在执行完一次循环之后，需要调用Sort函数重新将该节点按照从小到大的顺序插入到原有的节点排序序列中。而这也是保证前缀码中的任何两个都不互为前缀的重要机制。