1 Tree Reconstruction

Given the in-order and pre-order traversal of a binary tree, produce its post-order traversal. The node labels are unique.

Of course, some details about input and output are missing here.

Origin: an ACM ICPC problem (please, tell me which one; I forgot; around 1986); apparently reused later on several occasions
2 The Case of the Smithsonian Clocks

Two friends, whom we will call Arthur and Robert, were curators at the Museum of American History. Both were born in the month of May, one in 1932 and the other a year later.

Each was in charge of a beautiful antique clock. Both of the clocks worked pretty well, considering their ages, but one of them lost ten seconds an hour and the other gained ten seconds an hour.

On one bright day in January, the two friends set both clocks right at exactly 12 noon.

“You realize,” said Arthur, “that the clocks will start drifting apart, and they won’t be together again until—let’s see—why, on the very day you will be 47 years old. Am I right?”

Robert then made a short calculation. “That’s right!” he said.

Who is older, Arthur or Robert?

By Raymond Smullyan
3 Handshakes

An anthropologist, back from a field trip, was invited to a party with her husband. Always the scientist, she decided to observe and to make measurements on the curious American ritual of handshaking. Four other married couples attended the party. Whenever two people shook hands, she recorded that each of the two people shook hands one time. In that way, for all of them (including herself and her husband), she obtained the total number of times that each person shook hands. First, she noted two obvious facts. One didn’t shake hands with one’s own spouse, and one didn’t shake the same person’s hand twice. Then she looked at her list and observed an odd fact. If she didn’t count herself, the other nine people all shook hands a different number of times. That is, one person didn’t shake any hands, on shook only once, to one shaking the hands of all eight of the others.

Question: How many times did her husband shake hands?

Mr. Smith and his wife invited four other couples for a party. When everyone arrived, some of the people in the room shook hands with some of the others. Of course, nobody shook hands with their spouse and nobody shook hands with the same person twice.

After that, Mr. Smith asked everyone how many times they shook someone’s hand. He received different answers from everybody.

How many times did Mrs. Smith shake someone’s hand?

Folklore? I first learned it from Paul Halmos. See [1, 2].

Can you generalize this?
4 Ages

Observe the following conversation among two friends.

A : How old are your three children?
B : The product of the ages is 36.
A : Then I don’t know enough . . .
B : The sum of their ages equals my house number.
A : Then I still don’t know enough . . .
B : The oldest just became youth champion at the go club!
A : Ah, that helps.

How old are these children?

A variant

During a recent census, a man told the census taker that he had three children. When asked their ages, he replied, “The product of their ages is 72. The sum of their ages is the same as my house number.”

The census taker ran to the door and looked at the house number. “I still can’t tell,” she complained. The man replied, “Oh that’s right, I forgot to tell you that the oldest plays tennis.”

The census taker promptly wrote down the ages of the 3 children. How old are they?

Folklore

Can you generalize?
5 Problem A: Rare Order

A rare book collector recently discovered a book written in an unfamiliar language that used the same characters as the English language. The book contained a short index, but the ordering of the items in the index was different from what one would expect if the characters were ordered the same way as in the English alphabet. The collector tried to use the index to determine the ordering of characters (i.e., the collating sequence) of the strange alphabet, then gave up with frustration at the tedium of the task. You are to write a program to complete the collector’s work. In particular, your program will take a set of strings that has been sorted according to a particular collating sequence and determine what that sequence is.

**Input**  The input consists of an ordered list of strings of uppercase letters, one string per line. Each string contains at most 20 characters. The end of the list is signalled by a line that is the single character ‘#’. Not all letters are necessarily used, but the list will imply a complete ordering among those letters that are used. A sample input file is shown below.

```
XWY
ZX
ZXY
ZXW
YWWX
#
```

**Output**  Your output should be a single line containing uppercase letters in the order that specifies the collating sequence used to produce the input data file. Correct output for the input data file above is shown below.

```
XZYW
```

This problem was adapted from the 1989–90 ACM ICPC World Finals held in Washington D.C., USA.
6 What are the prices at 7–11

There’s a chain of stores in the United States called 7–11. They’re probably called this because they used to be open from 7 a.m. until 11 p.m., but now they’re usually open all the time. One day a customer arrived in one of these 7–11 shops and selected four items. He then approached the counter to pay for these items. The salesman took his calculator, pressed a few buttons and said, “The total price is $7.11.”

The customer tried a joke, “Why? Do I have to pay $7.11 because the name of your shop is 7–11?”

The salesman didn’t get the joke and answered, “Of course not! I just multiplied the prices of these four items and I have just given you the result!”

The customer was very surprised. “Why did you multiply these numbers? You should have added them to the total price!”

The salesman said, “Oh, yes, I’m sorry, I have a terrible headache and I pressed the wrong button!”

Then the salesman repeated all the calculations, i.e., added the prices of these four items, but to his and the customer’s great surprise, the total was still $7.11.

Now, the task is to find the prices of these four items!

See [2].

Can you generalize?
7 Wires and Switches

In Figure 1, a cable with three wires connects side $A$ to side $B$. On side $A$, the three wires are labeled 1, 2, and 3. On side $B$, wires 1 and 3 are connected to switch 3, and wire 2 is connected to switch 1.

In general, the cable contains $m$ wires ($1 \leq m \leq 90$), labeled 1 through $m$ on side $A$, and there are $m$ switches on side $B$, labeled 1 through $m$. Each wire is connected to exactly one of the switches. Each switch can be connected to zero or more wires.

![Figure 1: Cable with three wires and three switches](image)

**Measurements** Your program has to determine how the wires are connected to the switches by doing some measurements. Each switch can be made either conducting or non-conducting. Initially all switches are non-conducting. A wire can be tested on side $A$ with probe $P$: Lamp $L$ will light up if and only if the sensed wire is connected to a conducting switch.

Your program begins by reading one line with the number $m$ from standard input. It then can give three kinds of commands by writing a line to standard output. Each command starts with a single uppercase letter: T (Test a wire), C (Change a switch), and D (Done). Command T is followed by a wire label, C by a switch label, and D by a list whose $i$-th element is the label of the switch to which wire $i$ is connected.

After commands T and C, your program should read one line from standard input. Command T returns Y (Yes) when the wire’s switch is conducting (the lamp lights up), otherwise it returns N (No). Command C returns Y if the new switch state is conducting, and N otherwise. The effect of command C...
is to change the state of the switch (if it was conducting then it will be non-conducting afterwards and vice versa); the result is returned just for feedback.

Your program may give commands $T$ and $C$ mixed in any order. Finally, it gives command $D$ and terminates. Your program should give no more than nine hundred (900) commands in total.

**Example** Figure 2 presents an example conversation involving 8 commands relating to Figure 1.

<table>
<thead>
<tr>
<th>Standard Output</th>
<th>Standard Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>C 3</td>
<td>Y</td>
</tr>
<tr>
<td>T 1</td>
<td>Y</td>
</tr>
<tr>
<td>T 2</td>
<td>N</td>
</tr>
<tr>
<td>T 3</td>
<td>Y</td>
</tr>
<tr>
<td>C 3</td>
<td>N</td>
</tr>
<tr>
<td>C 2</td>
<td>Y</td>
</tr>
<tr>
<td>T 2</td>
<td>N</td>
</tr>
<tr>
<td>D 3 1 3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Example conversation

Origin: IOI 1995, Netherlands; see [5]; this problem arose from a real life situation.
8 Palindromes

Design an efficient algorithm that reads a sequence of characters and outputs the minimum number of characters to be inserted into this input sequence to make it a palindrome.

For example, for input ‘Ab3bc’ the output should be 2, because by inserting two characters a palindrome can be made (e.g., ‘Acb3bcA’), but no palindrome can be made by inserting fewer than two characters. Note, however, that it is not required to produce a witness.

Origin: IOI 2000, China
9 Find the Median

Given is an odd number $N$ of objects, all of distinct strength. The only way to compare strengths is through the function $\text{Med3}(a, b, c)$ that returns the object of median (middle) strength among three distinct objects:

$$\min\{a, b, c\} < \text{Med3}(a, b, c) < \max\{a, b, c\}$$

Design an efficient algorithm to determine the object of median strength among all given objects, using only function $\text{Med3}$.

As precondition, the number $N$ is bounded by $5 \leq N \leq 1499$, and, for each run, the number $C$ of calls to $\text{Med3}$ is bounded by $C \leq 7777$.

Origin: IOI 2000, China; see [6]
10 Coins and Balance

There are many problems involving coins and a balance. The following variant was communicated to me by Gerhard Woeginger.

Given are a number of coins in at most two different masses. That is, each coin either has mass $x$ or mass $y$, with $x \neq y$. The goal is to determine, using a balance, whether all coins have the same mass or not.

With eight coins, this is pretty straightforward to do in three weighings. (How?) The problem is to do it for ten coins in three weighings.

Next, you should generalize this. How many coins can you handle in four weighings? How many in five?
References


