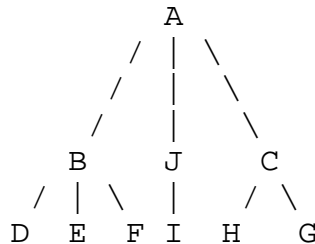# UCF Local Contest — September 1, 2012

Happy Hiking Grounds
filename: `hiking`

Arup likes hiking and, since he just taught trees in CS I, he has found a trail that is in the form of a tree. For simplicity, Arup always uses A to refer to the tree root. So, a sample trail might be:

```
                 A
               / | \
              /  |  \
             /   |   \
            B    J    C
          / | \  |   / \
         D  E  F I  H   G
```

Arup has decided to follow a Breadth First Search (BFS) to hike the trail and, for simplicity, Arup visits the children of a node in alphabetical order. The BFS traversal of the above sample tree is:

```
A  B  C  J  D  E  F  G  H  I
```

But Arup quickly realizes that to go from one child to a sibling, he has to go back to the parent again (since there is no direct path between a child and its sibling). So, for the above tree, Arup will actually walk as follows:

```
A-B-A-C-A-J-A-B-D-B-E-B-F-B-A-C-G-C-H-C-A-J-I-J-A
```

Note that Arup returns to the root at the end of his hike. Well, this is a lot more walking than Arup had expected from BFS but Arup doesn't mind since his wife (a medical doctor) has convinced him of the benefits of exercise and good eating habits!

The Problem:

Given a trail description (a tree), determine the total distance (cost) of the described Breadth First Search hike of the trail including the distance (cost) of backtracking at each step.

The Input:

The first input line contains a positive integer, t, indicating the number of trails (trees) to check. Each trail is defined by a series of paths that connect junctions and points of interest (each of which is represented by a single uppercase letter). On the following input line there will be an integer, n $(1 \leq n \leq 26)$, providing the number of tree nodes (junctions and points of interest); assume the nodes will be the first n uppercase letters. On each of the next $n-1$ input lines, there will be two uppercase letters, indicating that there is an edge between those two locations (first node being the parent of the second node), and a positive integer representing the length of that

edge (the distance between the two nodes); the three values on these input lines are separated from each other by a single space.

Arup will always start at the entrance, which is labeled as "A" (which also is guaranteed to appear in the input).

The Output:

At the beginning of each trail, output the message "`Trail #i:`" on a line by itself where *i* is the number of trail in the input (starting with 1). On the next output line, output the message "`BFS hike will cost b`" where *b* is the total distance of Arup's hike in his breadth first fashion (this must include the backtracking cost). Assume the total distance will fit in an integer.

Note that, at each location, Arup will always visit each of the next possible locations in alphabetical order. Leave a blank line after the output for each trail. Follow the format illustrated in Sample Output.

Sample Input:

```
2
3
A B 5
A C 3
8
A B 1
A C 7
B D 5
B E 3
B F 2
C G 4
C H 2
```

Sample Output:

```
Trail #1:
BFS hike will cost 16

Trail #2:
BFS hike will cost 64
```