

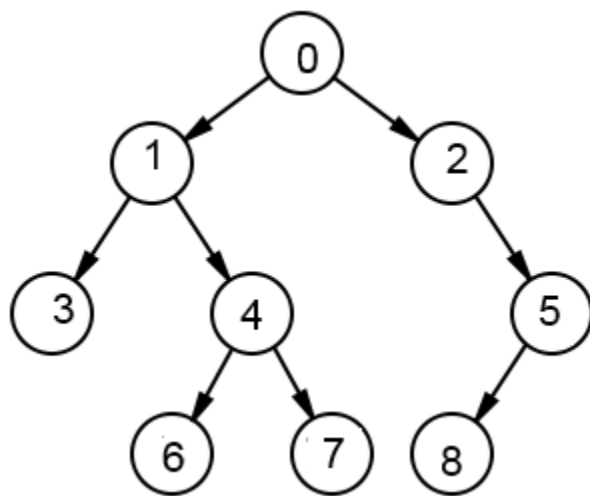
R&D Snickerlympics

Week 1

Overview

This week you are to explore working with graphs and trees (something I wish I could do in class but time won't suffice). Graphs and trees are extremely useful in practice so even if you don't succeed in implementing the algorithms below, please try and at least read more about them.

For all the problems you'll read a tree from the standard input (std::cin). The tree will be given by its adjacency list. On the first line of the standard input, the number of nodes n in the tree will be entered. On the next n lines, the indices of i th node's children will be given. If that node is a leaf, the line will be empty. For example, the tree to the left:



Will be represented by the following input:

```
9
1 2
3 4
5
<empty_line>
6 7
8
<empty_line>
<empty_line>
```

Problem 1. DFS

Walk the tree using depth first search and print all neighbors and children of every node. Children should be indented exactly 4 spaces to the right than their parents. No node should ever be repeated.

Example output on the tree above:

```
0
  1
    3
    4
      6
      7
  2
    5
      8
```

Problem 2. BFS

Walk the graph using breadth first search and print all neighbors and children of every node. Children should be indented exactly 4 spaces to the right than their parents. No node should ever be repeated.

Example output on the tree above:

```
0
  1
    2
      3
      4
      5
        6
        7
        8
```

UPLOAD SOLUTIONS TO IN MOODLE.