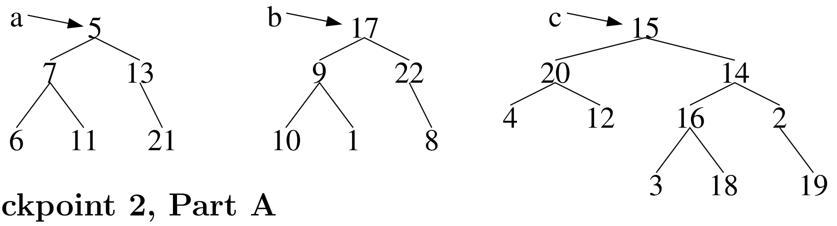
CSCI-1200 Data Structures — Spring 2020
Lab 11 — Hash Tables / Advanced Trees, Checkpoint 3

Pair up with one other student in your lab section and complete the exercises below.
Please raise your hand and ask for help from a TA or mentor if you get stuck.
In this problem we will compare the shape (not values) of binary trees.

```
class Node {
public:
  int value;
  Node* left;
  Node* right;
};
```



Checkpoint 2, Part A

estimate: 10-20 minutes

First, write a *recursive* function named `shape_match` that takes in two `Node` pointers and returns true if those trees have the same shape. For example, `shape_match(a,b)` returns true and `shape_match(a,c)` returns false.

Checkpoint 2, Part B

estimate: 10-20 minutes

Next, write a *recursive* function named `find_subtree_match` that takes a pointer to a large tree and a pointer to a pattern and returns a pointer to a node anywhere within the tree that matches the pattern, or NULL if there is no match. For example `find_subtree_match(c,a)` will return a pointer to the node storing 14.

Checkpoint 2, Part C *estimate: 20-30 minutes*

Now open up your laptop and test your code. In a new file, type up the `Node` class, your two functions from parts A & B, and write a simple tree print function for debugging. Now you can create the sample trees from the start of this handout in your `main` function. Then test and debug.

Checkpoint 2, Part D *estimate: unknown*

If time allows, consider the problem of finding the largest shape-matching subtree within two large trees. First draw a moderately complex example, then write this function.

To complete this checkpoint: When you have finished all of the problems on this worksheet *OR* you have been working on it for about 60 minutes (whichever comes first), put your names in the checkpoint queue and discuss your answers with your lab TA or mentor.