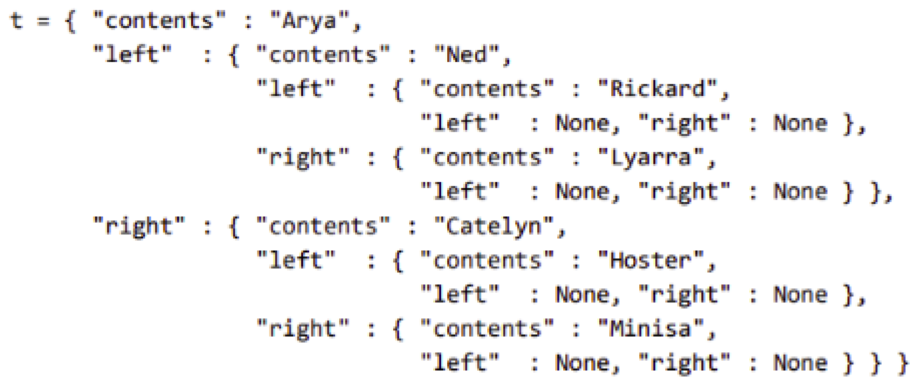
findParents(t, name)

Write the function findParents(t, name) which takes a genealogical family tree in the form of a binary tree (as discussed in class) and a string, the name of a person. If the person exists in the tree, the function returns a list containing their parents.

You'll need to use if statements to separate the four possible arrangements of parents. Return an empty list if the parents are unknown, a singleton list in the two cases where only one parent is known, and a two-element list if both parents are known. If the person does not exist in the tree, the function returns None. You are guaranteed that every name in the tree only shows up once (to avoid conflicting information).

For example, consider the following family tree (as shown in the course slides):



```
t = { "contents" : "Arya",
      "left"  : { "contents" : "Ned",
                  "left"  : { "contents" : "Rickard",
                              "left"  : None, "right" : None },
                  "right" : { "contents" : "Lyarra",
                              "left"  : None, "right" : None } },
      "right" : { "contents" : "Catelyn",
                  "left"  : { "contents" : "Hoster",
                              "left"  : None, "right" : None },
                  "right" : { "contents" : "Minisa",
                              "left"  : None, "right" : None } } }
```

If we called `findParents(t, "Ned")`, the function would return the list `["Rickard", "Lyarra"]`. If we called `findParents(t, "Rickard")`, the function would return [ ]. If we called `findParents(t, "Jon")`, the function would return None because "Jon" is not in the tree.

Hint 1: treat this like a recursive search problem (in fact, you might want to reference tree linear search and binary search if you get stuck). You'll need to make two base cases - one for when you find the person, one for when you reach a leaf or an empty tree. For this problem, it may be easier to set up the base case as an empty tree.

Hint 2: unlike recursive binary search, you need to check both branches of the tree to see if the person occurs in either branch. How can you combine their results? If one of the branches gives you a list, you've found the parents - return them immediately. If you get None, try again on the other branch. If both branches return None, then the name was not found and you should return None.

Note:

Given in the genealogical family tree, consider the first node as children and below nodes are parents of children nodes.

Tree Structure:

```
  Ned
       /   \
   Rickard  Lyarra
```

Tree Representation:

(INPUT)

```
tree = {"name": "Ned","left": {
        "name": "Rickard",
        "left": None,
        "right": None
    },
    "right": {
        "name": "Lyarra",
        "left": None,
        "right": None
    }
    }
```

Output:

```
["Rickard", "Lyarra"]
```