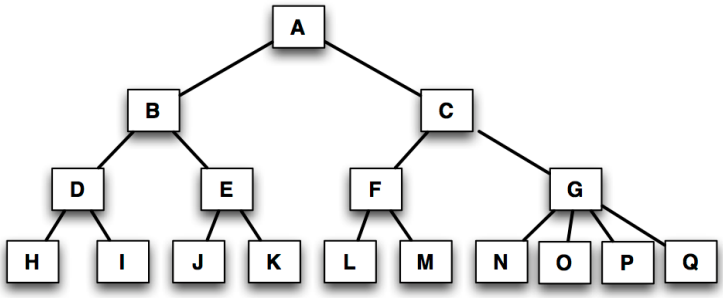
PROBLEM 6

Family trees have been around for a long time, even before computers were invented. The parents formed the root at the top, and below were their children, then grandchildren, and so on.

There are special words for parents, grandparents, uncles, aunts, and more. Below is a family tree, with "A" as the founding ancestor.



Click on the above image to see it at full size.

In this problem, we'll use the general term "cousin" as follows:

- zeroth cousin: If two nodes are siblings (have the same immediately preceding ancestor, such as nodes "H" and "I") they are zeroth cousins.
- first cousin: Children of zeroth cousins are first cousins.
- second cousin: Grandchildren of zeroth cousins are second cousins.
- In general, i'th cousins have a grandparent or ancestor that is i levels up from their parents.

PROBLEM 6-1 (1/1 point)

Match the pair with its definition:

" `D` " and " `G` "

- ○ zeroth cousin
- ◉ first cousin ✔
- ○ second cousin
- ○ third cousin

You have used 1 of 1 submissions

PROBLEM 6-2 (1/1 point)

Match the pair with its definition:

" `B` " and " `C` "

- (•) zeroth cousin ✔
- ( ) first cousin
- ( ) second cousin
- ( ) third cousin

You have used 1 of 1 submissions

PROBLEM 6-3 (1/1 point)

Match the pair with its definition:

"`I`" and "`Q`"

- ( ) zeroth cousin
- ( ) first cousin
- (•) second cousin ✔
- ( ) third cousin

You have used 1 of 1 submissions

Suppose two people, P1 and P2, are i'th cousins. Let C1 be a child of P1 and C2 be a child of P2. Then, C1 is an i'th cousin of P2, 1 removed, and C2 is an i'th cousin of P1, 1 removed.

Let G1 now be a child of C1. G1 is an i'th cousin of P2, 2 removed.

In general, the type of cousin (what 'i' is) is the shorter distance to the ancestor of two people, and the amount removed is the difference between the distance to the common ancestor.

Using this above definition for the next 3 problems, match the pairs with their relationship:

PROBLEM 6-4 (1/1 point)

"`D`" and "`G`"

- (•) first cousin 0 removed ✔
- ( ) first cousin 2 removed
- ( ) second cousin 1 removed

You have used 1 of 1 submissions

PROBLEM 6-5 (1/1 point)

"`D`" and "`M`"

- ( ) second cousin 0 removed
- ( ) zeroth cousin 2 removed
- (•) first cousin 1 removed ✔

You have used 1 of 1 submissions

PROBLEM 6-6 (1/1 point)

" `B` " and " `L` "

- second cousin 0 removed
- zeroth cousin 2 removed ✔
- second cousin 1 removed

You have used 1 of 1 submissions

PROBLEM 6-7 (20/20 points)

Consider the class definitions contained in FamilyTree.py. Class `Member` is a class that represents a single person in the family, and Class `Family` represents the whole family tree.

You are to write code for the method `cousin` of the class `Family` according to the docstring in `FamilyTree.py` and the definitions for degree removed and cousin type right before Problem 6-4.

Paste your entire definition of the `Family` class in the following box. You may assume that the class `Member` is defined for you. You should not alter `Member` in any way, but may alter any part of `Family` that you deem necessary.

Please try the problem first without looking at the hints.

Hints

I'm really stuck!

```
# Paste your entire definition of the Family class in the following box.
class Family(object):
    def __init__(self, founder):
        """
        Initialize with string of name of oldest ancestor

        Keyword arguments:
        founder -- string of name of oldest ancestor
        """

        self.names_to_nodes = {}
        self.root = Member(founder)
        self.names_to_nodes[founder] = self.root

    def set_children(self, mother, list_of_children):
        """
```

Correct

Test results

Hide output

CORRECT

Test: Relationship Test 1

Output:

```
t, r = f.cousin("b", "c")
'b' is a zeroth cousin 0 removed from 'c'

Test completed
```

Test: Relationship Test 2

Output:

```
t, r = f.cousin("d", "f")
'd' is a first cousin 0 removed from 'f'

Test completed
```

Test: Relationship Test 3

Output:

```
t, r = f.cousin("i", "n")
'i' is a second cousin 0 removed from 'n'

Test completed
```

Test: Relationship Test 4

Output:

```
t, r = f.cousin("q", "e")
'q' is a first cousin 1 removed from 'e'

Test completed
```

Test: Relationship Test 5

Output:

```
t, r = f.cousin("h", "c")
'h' is a zeroth cousin 2 removed from 'c'

Test completed
```

Test: Relationship Test 6

Output:

```
t, r = f.cousin("h", "a")
'h' is a non cousin 3 removed from 'a'

Test completed
```

Test: Relationship Test 7

Output:

```
t, r = f.cousin("h", "h")
'h' is a non cousin 0 removed from 'h'

Test completed
```

Test: Relationship Test 8

Output:

```
t, r = f.cousin("a", "a")
'a' is a non cousin 0 removed from 'a'

Test completed
```

Test: Relationship Test Randomized 1

Output:

```
t, r = f.cousin("b", "g")
'b' is a zeroth cousin 1 removed from 'g'

Test completed
```

Test: Relationship Test Randomized 2

Output:

```
t, r = f.cousin("a", "k")
'a' is a non cousin 3 removed from 'k'

Test completed
```

Test: Relationship Test Randomized 3

Output:

```
t, r = f.cousin("e", "h")
'e' is a zeroth cousin 1 removed from 'h'

Test completed
```

Test: Relationship Test Randomized 4

Output:

```
t, r = f.cousin("a", "i")
'a' is a non cousin 3 removed from 'i'

Test completed
```

Test: Relationship Test Randomized 5

Output:

```
t, r = f.cousin("g", "m")
'g' is a zeroth cousin 1 removed from 'm'

Test completed
```

Test: Relationship Test Randomized 6

Output:

```
t, r = f.cousin("j", "g")
'j' is a first cousin 1 removed from 'g'

Test completed
```

Test: Relationship Test Randomized 7

Output:

```
t, r = f.cousin("h", "o")
'h' is a second cousin 0 removed from 'o'

Test completed
```

Test: Relationship Test Randomized 8

Output:

```
t, r = f.cousin("o", "o")
'o' is a non cousin 0 removed from 'o'

Test completed
```

Hide output

Check Save *You have used 1 of 10 submissions*

literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

edX Blog

Donate to edX

Jobs at edX

Google+