Review Q & A - Mar. 10

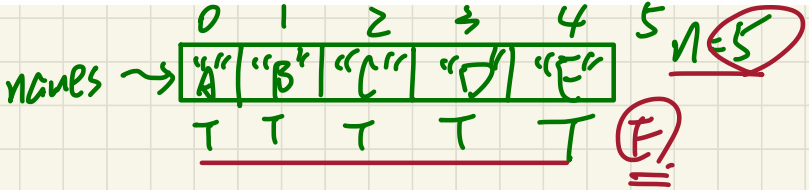
Written Test

Asymptotic Analysis
Instantiating Generics

```
① boolean foundEmptyString = false;
② int i = 0;
③ while (!foundEmptyString && i < names.length) {
④   if (names[i].length() == 0) {
⑤     /* set flag for early exit */
⑤     foundEmptyString = true;
⑤   }
⑥   i = i + 1;
  }
```

worst case is when the "" is never found

→ # iterations = names.length = n



n = 5

① 1

② 1

③ (n+1) · 4

④, ⑤, ⑥ 6 · n

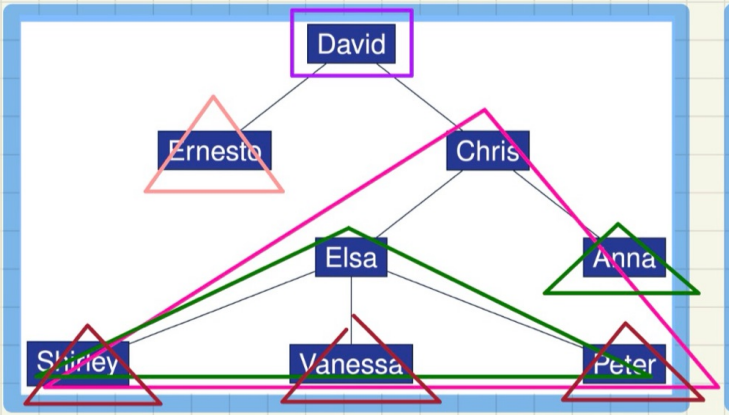
iterations : n

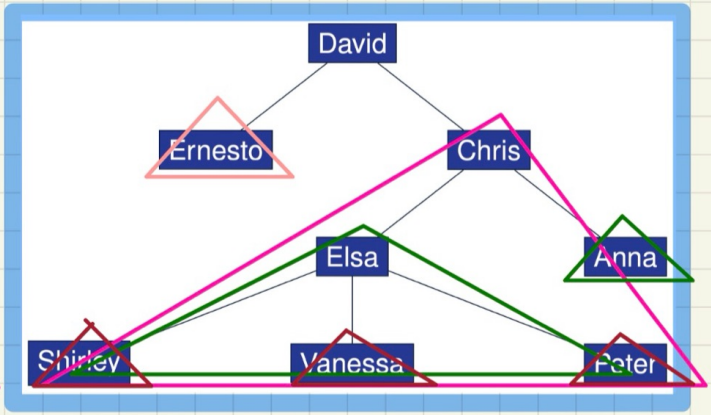
times while cond. is evaluated : n+1 (F)

1 + 1 + (4n+4) + 6n

= 10n + 6

General Tree Traversals: Pre-Order vs. Post-Order





Pre-Order Traversal from the Root

David Ernesto po(E) Chris Elsa S V P po(Elsa) po(Chris) Anna po(Anna)

Post-Order Traversal from the Root

Ernesto po(E) S V P Elsa po(Elsa) Anna po(A) Chris po(C) David

a1.length == n a2.length == m

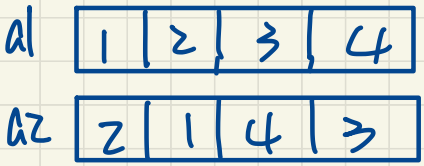
```
boolean contentsMatch(int[] a1, int[] a2){
    for(int i=0; i< a1.length; i++){
        for(int j=0; j< a2.length; j++){
            // check to see if a2[j] == a1[i]   O(1)
        }
    }
    for(int i=0; i< a2.length; i++){
        for(int j=0; j< a1.length; j++){
            // check to see if a1[j] == a2[i]   O(1)
        }
    }
}
```

n · m

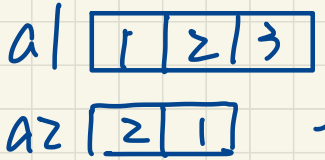
$O(n \cdot m \cdot 1 + m \cdot n \cdot 1)$

$= O(2 \cdot n \cdot m)$

$= O(n \cdot m)$ m · n



true



false

$S1 = S2 \iff S_1 \subseteq S_2 \wedge S_2 \subseteq S_1$

highest power

$5n^2 + 3n \cdot \log n + 2n + 5$ is $O(n^2)$ $[c = 15, n_0 = 1]$

$f(n)$ $g(n)$

Proof

choose $C = |5| + |3| + |2| + |5| = 15$

$n_0 = 1$

Verify: $f(1) \leq C \cdot g(1)$

$5 \cdot 1^2 + 3 \cdot 1 \cdot \log 1 + 2 \cdot 1 + 5$

$= 12$

$\leq 15 \cdot 1^2 = 15$

```
public class MyClass<G, H, I> {
    private G[] a;
    public G m1 (I p1, H p2) {
        /* details of implementation omitted */
        return null;
    }
    public void m2 (H p1,  G p2) {
        /* details of implementation omitted */
    }
}
```

```
public class MyClass<G, H, I> {
    private G[] a;
    public G m1 (I p1, H p2) {
        /* details of implementation omitted */
        return null;
    }
    public void m2 (H p1,  G p2) {
        /* details of implementation omitted */
    }
}
```

Now consider the following declarations from another class (which intends to use the above generic class):

```
Boolean b;
String s;
Integer i;
MyClass<Integer, String, Boolean> obj1;
MyClass<Boolean, String, Integer> obj2;
```

s = obj1.m2 ("alan", 5);	Choose...
b = obj2.m2 ("alan", **false**);	Choose...
obj1.m1 ("alan", 5);	Choose...
obj2.m2 ("alan", **false**);	Choose...
obj1.m2 (5, "alan");	Choose...
obj2.m2 (**false**, "alan");	Choose...
i = obj1.m1 (**true**, "mark");	Choose...
b = obj2.m1 (3, "mark");	Choose...
b = obj1.m1 (**true**, "mark");	Choose...
i = obj2.m1 (3, "mark");	Choose...

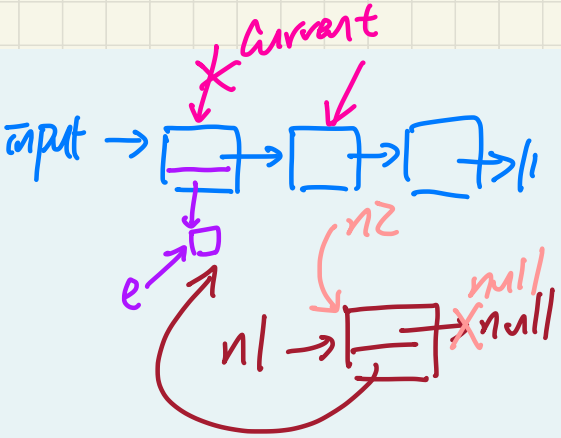
Consider the following method which intends to reverse the input chain of nodes:

```
public Node<String> reverseOf(Node<String> input) {
    Node<String> n2 = null;
    Node<String> current = input;
    while(current != null) {
        String e = current.getElement();
        Node<String> n1 = new Node<>(e, null);

        /* missing some line(s) of code */

        current = current.getNext();
    }
    return n2;
}
```

n1.setNext(n2);
n2 = n1;



In the above method, some line or lines of code are missing (from where the comment is) in order for the implementation to be correct. Choose **the** line or **all** lines that are needed.

- [ ] n1.setNext(n2);
- [ ] n2 = n1;
- [ ] n2.setNext(n1);
- [ ] n1.setNext(current);
- [ ] n1 = n2;
- [ ] n2 = current;

Your answer is incorrect.

The correct answers are:

n1.setNext(n2);,

n2 = n1;