1 Runtime Analysis

Find out the Big-O runtime and Tight-O runtime of the following methods:

1.
```
int a = 0, b = 0;
for(i = 0; i < N; i++) {   ← N times
    a = a + rand();
}

for(j = 0; j < M; j++) {   ← M times
    b = b + rand();
}
```

$\Theta(N+M)$

$O(N+M)$

2.
```
int a = 0;
for(i = 0; i < N; i++) {   ← N times
    for(j = N; j > i; j--) {   ← decrement by 1 each time.
        a = a + i + j;
    }
}
```

$N + N-1 + N-2 + N-3 \dots + 2 + 1 + 0$

$= N * (N+1) / 2$

$\Theta(\frac{1}{2}N^2 + \frac{1}{2}N)$

$O(N^2)$

3.
```
int i, j, k = 0;
for(i = N/2; i < N; i++) {   ← N/2
    for(j = 2; j <= N; j = j*2) {   ← log N
        k = k + N/2;
    }
}
```

$$\boxed{\begin{aligned} & j = 2, 2^2, 2^3, 2^4 \dots \\ & N = 2^{\text{times}} \\ & \text{times} = \log(N) \end{aligned}}$$

$\Theta(\frac{N}{2} \cdot \log N)$

$O(N \log N)$

4.
```
int a = 0, i = N;
while(i > 0) {
    a += i;
    i /= 2;
}
```

i :

$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \dots \rightarrow \frac{N}{2^{\text{times}}}$

$\boxed{\text{while}(i > 0)}$

$\frac{N}{2^{\text{times}}} = 1$ ↵

$N = 2^{\text{times}}$

$\text{times} = \log(N)$

$\Theta(\log N)$

$O(\log N)$

2 Review

Toby wants to rule the world with an army of cats. Each cat may or may not have one parent, and may or may not have 'kitties'. Each cat that has a parent is a 'kitty' of that parent. But after implementing copyCat, which creates a copy of a cat and its descendants, he realizes the function contains a bug.

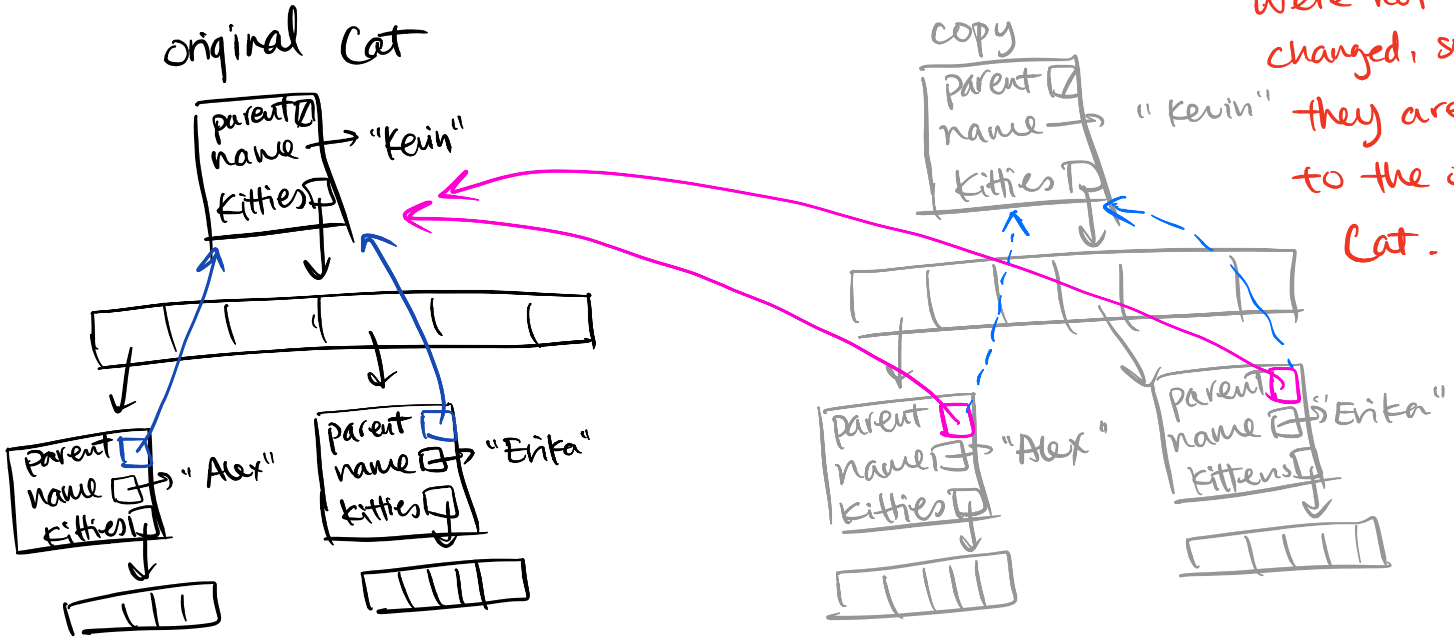
```
public class Cat {
    private Cat parent;
    private ArrayList<Cat> kitties;
    private String name;

    public Cat(Cat parent, String name) {
        this.name = name;
        this.kitties = new ArrayList<Cat>();
        this.parent = parent;
    }

    public Cat copyCat() {
        Cat copy = new Cat(this.parent, this.name);
        for (int i = 0; i < this.kitties.size(); i += 1) {
            copy.kitties.add(this.kitties.get(i).copyCat());
        }
        return copy;
    }
}
```

What's wrong with his Cat class? Drawing a box and pointer diagram may help!

the parent pointers of the kitties were not changed, so they are pointing to the original Cat.


original Cat
parent
name
kitties
"Kevin"
"Alex"
"Erika"
copy
"Kevin"
"Alex"
"Erika"
kittens