

1 Warmup

Given the following method on a sorted array, what is the worst-case runtime?.

```
1 public static int f1(int i, int[] numList) {
2     for (int j = 0; j < numList.length; j++) {
3         if (numList[j] == i) {
4             return j;
5         }
6     }
7     return -1;
8 }
```

Rewrite this method to improve the runtime. What is the runtime of your new method?

2 You wanna hang out this Spring '15? Asymptotes!

For each of the pieces of code below, give the runtime in θ notation as a function of the given parameters. Let $f(x)$ be a function that runs in time linear to the size of its input x .

```
1 public static void f1(int n) {
2     if (n == 0) {return;}
3     f1(n/2);
4     f(n);
5     f1(n/2);
6 }
```

```
1 public static void f2(int n) {
2     if (n == 0) {return;}
3     f2(n-1);
4     f(17);
5     f2(n-1);
6 }
```

```
1 public static void f3(int n, int m) {
2     if (m <= 0) {
3         return;
4     } else {
5         for (int i = 0; i < n; i +=1) {
6             f3(n, m - 1);
7         }
8     }
9 }
```

3 It's Fall '16 And I'm Still Doing Asymptotics

In the following, assume that `cnst()` is a constant time function call. When giving bounds, give bounds that are as tight and simple as possible.

1. Give best- and worst-case runtime bounds for the call `foo2(N,N)` as a function of N

```
1 public static void foo2(int i, int N) {
2     if (i==0) {return;}
3     for (int j = 0; j < i; j = j+1) {
4         cnst();
5     }
6     if (i > N/2) {
7         foo2(i-1, N);
8     } else {
9         foo2(i/2, N) + foo2(i/2, N);
10    }
11 }
```

Best case: $\theta(\quad)$ Worst case: $\theta(\quad)$

2. True or false: if $f(N) \in O(N)$ and $g(N) \in O(N^2)$, and both functions are non-negative, then $|g(N) - f(N)| \in \Omega(N)$. if true, explain why; otherwise, give a counterexample.
3. True or false: if $f(N) \in \theta(N)$ and $g(N) \in \theta(N^2)$, and both functions are non-negative, then $|g(N) - f(N)| \in \Omega(N)$. If true, explain why; otherwise give a counterexample.

4. What is the running time of the following algorithm, as a function of the parameter r ?

```
1 /** Assumes that VALS is a square array, and that 0 <= R, C <
2     vals.length. */
3 double best(double vals[][], int r, int c) {
4     if (r == 0) {
5         return vals[r][c];
6     }
7     double v = best(vals, r-1, c);
8     if (c > 0) {
9         v = Math.max(v, best(vals, r-1, c-1));
10    }
11    if (c < vals[r].length - 1) {
12        v = Math.max(v, best(vals, r-1, c+1));
13    }
14    return v + vals[r][c];
15 }
```

Answer:

4 Fall '16: I'm more than just a runtime

1. Your friend, a budding politician, meets several hundred people a day and places their names onto the front of an ArrayList. Once there, he never removes a name, but he sometimes looks through the list to see the order in which he met people. At least one aspect of his procedures is slower than it could be. Describe and justify a small change in your friend's use of the data structure that would improve runtime without changing the data structure involved.

Now describe and justify a change in the data structure that would improve runtime without requiring a change in actions taken.

5 (Most 61B Problems) \in O(These Problems)

1. Give a tight θ bound on the running time.

```
1 public int f1(n):
2     if (n == 1) {return 0;}
3     if (n is even) {
4         return f3(n/2);
5     } else {
6         return f3(n+1);
7     }
```

Answer: $\theta()$

2. Give a tight θ bound on the running time, where process() is a method that runs in $\theta(n \log n)$

```
1 function f2(n):
2     if (n = 1) {return 1;}
3     int a = f2(n/2);
4     int b = f2(n/2);
5     x = process(a, b)
6     return x;
```

Answer: $\theta()$

3. True or False: If $f(n) = O(g(n))$, then $2^{f(n)} = O(2^{g(n)})$