6 Asymptotics

(18 Points)

Give the best case runtime and worst case runtime for the functions below.

(a)

```java
public static void a(int N) {
    for (int i = 1; i < N; i += 1) {
        for (int j = 1; j < i * i; j += 1) {
            System.out.println("a");
        }
    }
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

(b)

```java
public static void b(int N) {
    if (N <= 1) {
        return;
    }
    for (int i = 0; i < N; i += 2) {
        System.out.println("b");
    }
    b(N / 3);
    b(N / 3);
    b(N / 3);
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

(c)

```
public static void c(int N) {
    Random rand = new Random();
    for (int i = 1; i < N; i *= 2) {
        for (int j = 0; j != rand.nextInt(0, i); j += 1) {
            System.out.println("c");
        }
    }
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

(d)

```
public static void d(int[] arr) {
    int N = arr.length;
    BSTSet<Integer> tree = new BSTSet<>();
    /* Assume that BST implements a binary search tree
    * with no self-balancing optimizations, as seen in lecture */
    for(int i = 0; i < N; i += 1) {
        tree.insert(arr[i]);
    }
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

(e)

```
public static void e(int N) {
    if (N <= 0) { return; }
    if (N % 2 == 0) { return; }
    e(N - 1);
    e(N - 2);
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

(f)

```
public static void f(int N) {
    if (N == 1) { return; }
    if (isPowerOfTwo(N)) {
        f(N / 2);
    } else {
        f(N - 1);
    }
}
public static boolean isPowerOfTwo(int N) {
    if (N == 1) {
        return true;
    } else if (N % 2 != 0 || N == 0) {
        return false;
    } else {
        return isPowerOfTwo(N / 2);
    }
}
```

Best case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above

Worst case:

○ $\Theta(1)$ ○ $\Theta(\log(\log N))$ ○ $\Theta(\log N)$ ○ $\Theta((\log N)^2)$ ○ $\Theta(N)$ ○ $\Theta(N \log N)$
○ $\Theta(N^2)$ ○ $\Theta(N^2 \log N)$ ○ $\Theta(N^3)$ ○ $\Theta(N^3 \log N)$ ○ $\Theta(N^4)$ ○ $\Theta(N^4 \log N)$
○ Worse than $\Theta(N^4 \log N)$ ○ Never terminates (infinite loop) ○ None of the above