2   Awesometotics

(22 Points)

For parts (a) through (d), determine the asymptotic behavior, with respect to N, of each scenario. Answer in Θ notation.

Example: A square of side length N is constructed. What is the area of that square?

$$\boxed{\Theta(N^2)}$$

(a) An `ArrayList` is implemented with additive resizing, then N elements are added to it. What is the maximum number of unused indices in the underlying array?

$$\boxed{}$$

(b) An `ArrayList` is implemented with multiplicative resizing, then N elements are added to it. What is the maximum number of unused indices in the underlying array?

$$\boxed{}$$

(c) You create a weighted quick union **without** path compression, containing N elements. You then perform $\lfloor\sqrt{N}\rfloor$ calls to `union`. You look at the underlying array and take the sum of the absolute value of all elements. What is the smallest and largest possible sum?

For example, if the underlying array was $[-4, 2, 0, 0]$, the sum would be $|-4| + |2| + |0| + |0| = 6$.

Note: $\lfloor x \rfloor$ is the largest integer less than or equal to x.

Smallest sum: $\boxed{}$

Largest sum: $\boxed{}$

(d) A list of positive integers (not necessarily of length N) is created such that its elements sum to N^2. All elements are inserted, one at a time, into a min-heap. What are the best-case and worst-case runtimes for all insertions to complete?

Best case: $\boxed{}$

Worst case: $\boxed{}$

For parts (e) and (f), give the best-case and worst-case runtimes for the functions below.

(e) **public static void** f1**(int** N) {

```
    for (int i = 0; i < N; i += 1) {
        for (int j = i; j < N; j += 1) {
            if (j % 2 == 0) {
                i += 1;
            }
            System.out.println("naming things is hard - Dylan");
        }
    }
}
```

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** f2**(int** N) {

```
    if (N < 2) {
        return;
    }
    f2(N/2);
    for (int i = 0; i < Math.sqrt(N); i += 1) {
        System.out.println("Aniruth Narayanan aka the Baller");
    }
    f2(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