3 Asymptotics

(12 Points)

Give the asymptotic runtimes for the functions below in Θ notation. Here are some common arithmetic rules that may be of use:

- $\sqrt{n} = n^{(0.5)}$
- $\log_a(b) - \log_a(c) = \log_a(b/c)$
- $\log_a(b) + \log_a(c) = \log_a(b \cdot c)$
- $\log_a(b^n) = n \log_a(b)$
- $\log_a(b) = \log_c(b) / \log_c(a)$
- $a^{\log_a(b)} = b$

(a)
```
public static void q1(int N) {
    for (int i = 1; i < N; i += 1) {
        for (int j = 1; j < N; j *= 2) { System.out.println("ben"); }
    }
}
```

$\Theta($ $)$

(b)
```
public static void q2(int N) {
    if (N <= 1) { return; }
    if ((new Random()).nextInt() % 2 == 0) {
        for (int i = 0; i < N / 2; i += 1) { System.out.println("beben"); }
        q2(N - 1);
    } else {
        q2(N - 4);
    }
}
```

Best Case

$\Theta($ $)$

Worst Case

$\Theta($ $)$

(c)
```
double q3(double N) {
    if (N <= 10) {
        return 1;
    }
    double C = N;
    while (C > 1) {
        System.out.println("aniruth");
        C = C / 2;
    }
    return q3(Math.sqrt(N));
}
```

$\Theta($ $)$

(d)
```
double q4(double N) {
    if (N <= 10) { return 1; }
    double C = N;
    while (C > 1) {
        System.out.println("anianiruth");
        C = C / 2;
    }
    return q4(Math.sqrt(N)) + q4(Math.sqrt(N));
}
```

(1) What is the height of the recursive tree in terms of N? We will define this as h.

$h = \Theta($ $)$

(2) What is the asymptotic runtime in terms of N and h?

$\Theta($ $)$