CS 61B
Spring 2026

Asymptotics I, Disjoint Sets
Exam-Level Discussion 06: February 23, 2026

1 xStep

Analyze the following loops and determine the asymptotic runtime of each using big Theta notation with respect to N. Assume that **System.out.println(...)** runs in constant time.

```
int x = 7;
while (x < N + 14) {
  System.out.println("tidal wave");
  x++;
}
```

Runtime: $\Theta($ ____________ $)$

```
for (int x = 1; x < N; x++) {
  for (int y = 1; y < N; y++) {
    System.out.println("amethyst");
  }
}
```

Runtime: $\Theta($ ____________ $)$

```
for (int x = 1; x < N; x *= 2) {
  for (int y = 1; y < N; y++) {
    System.out.println("flamewall");
  }
}
```

Runtime: $\Theta($ ____________ $)$

```
for (int x = 2; x < N; x += 2) {
  for (int y = 1; y < 1000000; y) {
    System.out.println("anathema");
  }
}
```

Runtime: $\Theta($ ____________ $)$

```
for (int x = 3; x < N * N * N; x *= 3) {
  System.out.println("nullscapes");
}
```

Runtime: $\Theta($ ____________ $)$

```
for (int x = 6; x < N; x += 6) {
  for (int y = x; y < N; y += 6) {
    System.out.println("grief");
  }
}
```

Runtime: $\Theta($ ____________ $)$

2 I am Speed

For each code block below, fill in the blank(s) so that the function has the desired runtime. Do not use any commas. If the answer is impossible, just write “impossible” in the blank. Assume that `System.out.println` runs in constant time. You may use Java’s `Math.pow(x, y)` to raise `x` to the power of `y`.

(a) Desired Runtime: $\Theta(N)$

```
public static void f1(int N) {
    for (int i = 1; i < N; ____________________){
        System.out.println("hi Dom");
    }
}
```

(b) Desired Runtime: $\Theta(\log N)$

```
public static void f2(int N) {
    for (int i = 1; i < N; ____________________) {
        System.out.println("howdy Ergun");
    }
}
```

(c) Desired Runtime: $\Theta(1)$

```
public static void f3(int N) {
    for (int i = 1; ____________________; i += 1) {
        System.out.println("hello Anniyat");
    }
}
```

(d) Desired Runtime: $\Theta(2^N)$

This one is tricky! *Hint: think about the dominating term in* $1 + 2 + 4 + 8 + ... + f(N)$

```
public static void f4(int N) {
    for (int i = 1; ____________________; i *= 2) {
        for (int j = 0; j < i; j += 1) {
            System.out.println("what's up Alyssa");
        }
    }
}
```

3 Disjoint Sets

For each of the arrays below, write whether this could be the array representation of a weighted quick union with path compression and explain your reasoning. **Break ties by choosing the smaller integer to be the root.**

(a)

1	2	3	0	1	1	1	4	4	5

(b)

9	0	0	0	0	9	9	9	−10

(c)

1	2	3	4	5	6	7	8	9	−10

(d)

−10	0	0	0	0	1	1	1	6	2

(e)

−10	0	0	0	0	1	1	1	6	8

(f)

−7	0	0	1	1	3	3	−3	7	7