

Algorithmic Analysis

Discussion 07

Announcements

- HW 4 due Tuesday 03/01
- Enigma due Friday 03/04

Review

Cost

Time Complexity (Time Cost) - How long does it take to run this program if we feed it certain input?

Space Complexity (Spatial Cost) - How much space does this program take to run on our computer?

Worksheet

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$
- C. $O(N)$ vs. $O(N^2)$
- D. $\Theta(N^2)$ vs. $O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ → $\Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$
- C. $O(N)$ vs. $O(N^2)$
- D. $\Theta(N^2)$ vs. $O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ $\rightarrow \Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$ \rightarrow Neither
- C. $O(N)$ vs. $O(N^2)$
- D. $\Theta(N^2)$ vs. $O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ $\rightarrow \Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$ \rightarrow Neither
- C. $O(N)$ vs. $O(N^2)$ \rightarrow Neither
- D. $\Theta(N^2)$ vs. $O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ $\rightarrow \Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$ \rightarrow Neither
- C. $O(N)$ vs. $O(N^2)$ \rightarrow Neither
- D. $\Theta(N^2)$ vs. $O(\log N)$ $\rightarrow O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ $\rightarrow \Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$ \rightarrow Neither
- C. $O(N)$ vs. $O(N^2)$ \rightarrow Neither
- D. $\Theta(N^2)$ vs. $O(\log N)$ $\rightarrow O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$ \rightarrow Neither

Why do we need to assume that N is large?

1 What's Faster?

Which is faster?

- A. $\Theta(N)$ vs. $\Theta(N^2)$ $\rightarrow \Theta(N)$
- B. $\Omega(N)$ vs. $\Omega(N^2)$ \rightarrow Neither
- C. $O(N)$ vs. $O(N^2)$ \rightarrow Neither
- D. $\Theta(N^2)$ vs. $O(\log N)$ $\rightarrow O(\log N)$
- E. $O(N \log N)$ vs. $\Omega(N \log N)$ \rightarrow Neither

Why do we need to assume that N is large?

Asymptotic bounds only make sense as N gets large because it allows us to disregard constant factors and lower order terms.

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in _ (g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in _ (g(x) = x^5)$$

$$f(x) = \log(x) \in _ (g(x) = 5x)$$

$$f(x) = e^x \in _ (g(x) = x^5)$$

$$f(x) = \log(5^x) \in _ (g(x) = x)$$

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in \Theta(g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in _ (g(x) = x^5)$$

$$f(x) = \log(x) \in _ (g(x) = 5x)$$

$$f(x) = e^x \in _ (g(x) = x^5)$$

$$f(x) = \log(5^x) \in _ (g(x) = x)$$

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in \Theta(g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in O(g(x) = x^5)$$

$$f(x) = \log(x) \in \Omega(g(x) = 5x)$$

$$f(x) = e^x \in \Omega(g(x) = x^5)$$

$$f(x) = \log(5^x) \in \Omega(g(x) = x)$$

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in \Theta(g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in O(g(x) = x^5)$$

$$f(x) = \log(x) \in O(g(x) = 5x)$$

$$f(x) = e^x \in \Omega(g(x) = x^5)$$

$$f(x) = \log(5^x) \in \Omega(g(x) = x)$$

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in \Theta(g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in O(g(x) = x^5)$$

$$f(x) = \log(x) \in O(g(x) = 5x)$$

$$f(x) = e^x \in \Omega(g(x) = x^5)$$

$$f(x) = \log(5^x) \in \Theta(g(x) = x)$$

2 Basic Algorithmic Analysis

$$f(x) = x^2 \in \Theta(g(x) = x^2 + x)$$

$$f(x) = 5000000x^3 \in O(g(x) = x^5)$$

$$f(x) = \log(x) \in O(g(x) = 5x)$$

$$f(x) = e^x \in \Omega(g(x) = x^5)$$

$$f(x) = \log(5^x) \in \Theta(g(x) = x)$$

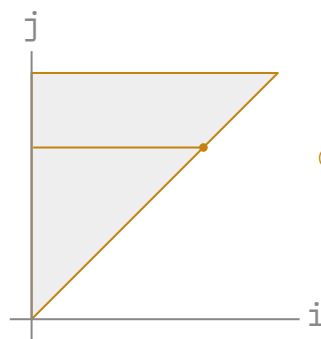
3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) {
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) {
9         constant(k);
10    }
11 }
```

3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) {
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) {
9         constant(k);
10    }
11 }
```

when $i = x$, the inner loop does x work



$$\begin{aligned} & \Theta(1) + \Theta(2) + \dots + \Theta(N) \\ &= \Theta(N * (N + 1) / 2) \\ &= \Theta(N^2) \end{aligned}$$

3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) { //  $\Theta(N^2)$ 
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) {
9         constant(k);
10    }
11 }
```

3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) { //  $\Theta(N^2)$ 
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) {
9         constant(k);
10    }
11 }
```

1 work for each of the N steps

3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) { //  $\Theta(N^2)$ 
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) { //  $\Theta(N)$ 
9         constant(k);
10    }
11 }
```


3A Practice with Runtime

```
1 public static void bars(int n) {
2     for (int i = 0; i < n; i += 1) { //  $\Theta(N^2)$ 
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) { //  $\Theta(N)$ 
9         constant(k);
10    }
11 }
```

$$\Theta(N^2 + N) = \Theta(N^2)$$

3A Practice with Runtime

```
1 public static void bars(int n) { //  $\Theta(N^2)$ 
2     for (int i = 0; i < n; i += 1) { //  $\Theta(N^2)$ 
3         for (int j = 0; j < i; j += 1) {
4             System.out.println(i + j);
5         }
6     }
7
8     for (int k = 0; k < n; k += 1) { //  $\Theta(N)$ 
9         constant(k);
10    }
11 }
```

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) {
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) { <- This whole thing is independent of N!
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) { //  $\Theta(1)$ 
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) { //  $\Theta(1)$ 
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j); // We know this is constant time now
15         }
16     }
17 }
```

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) { //  $\Theta(1)$ 
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
```

```
10
11 public static void barsRearranged(int n) {
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```

$$\begin{aligned} & \Theta(1) + \Theta(2) + \dots + \Theta(N/2) + \Theta(N) \\ & = \sum_{i=0}^{\log N} 2^i \\ & = \Theta(2^{(\log N + 1)} - 1) \\ & = \Theta(N) \end{aligned}$$

3B Practice with Runtime

```
1 public static void cowsGo(int n) {
2     for (int i = 0; i < 100; i += 1) { //  $\Theta(1)$ 
3         for (int j = 0; j < i; j += 1) {
4             for (int k = 0; k < j; k += 1) {
5                 System.out.println("moove");
6             }
7         }
8     }
9 }
10
11 public static void barsRearranged(int n) { //  $\Theta(N)$ 
12     for (int i = 1; i <= n; i *= 2) {
13         for (int j = 0; j < i; j += 1) {
14             cowsGo(j);
15         }
16     }
17 }
```