CSE 250: Asymptotic Analysis

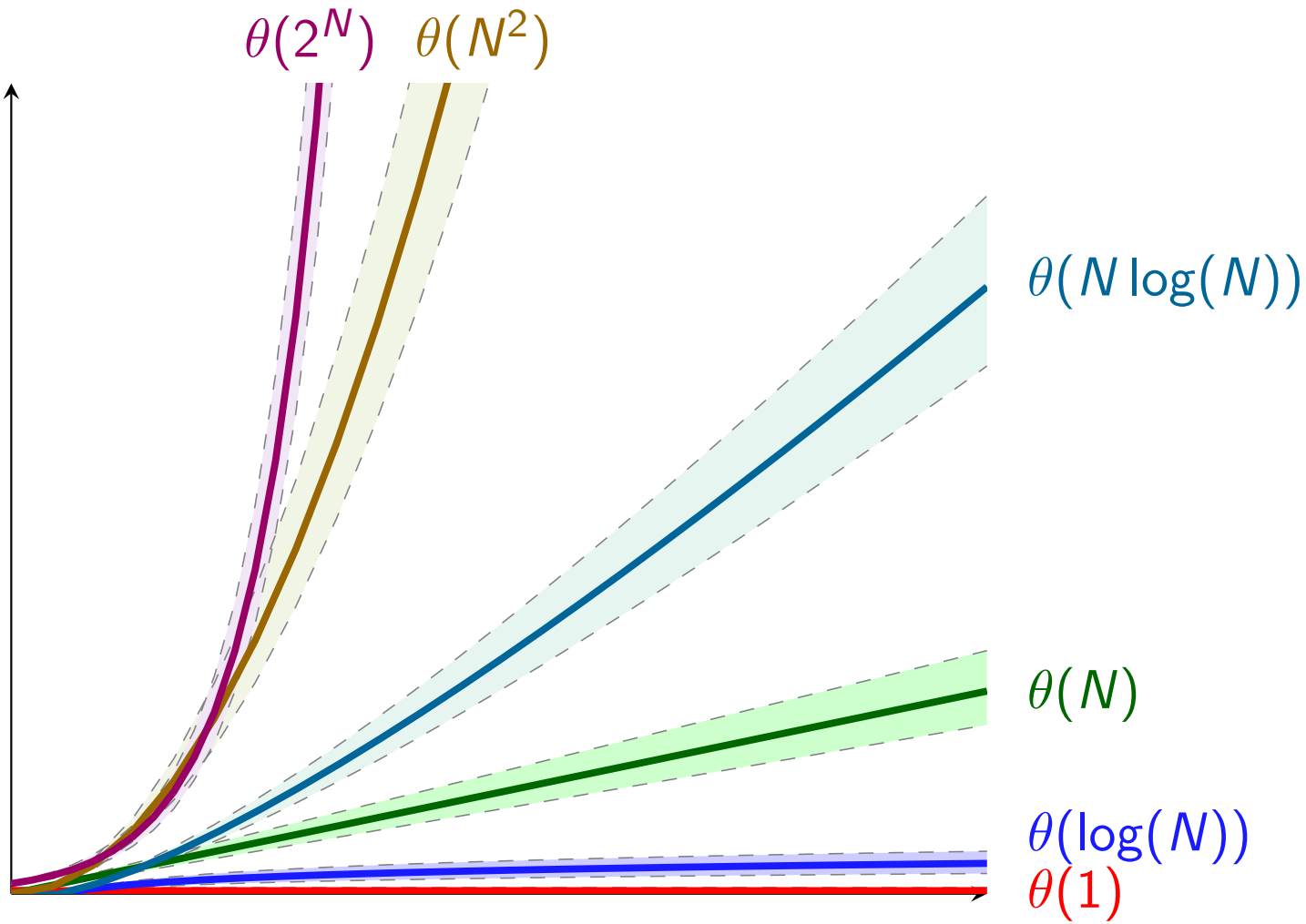
Lecture 6

Sept 11, 2023

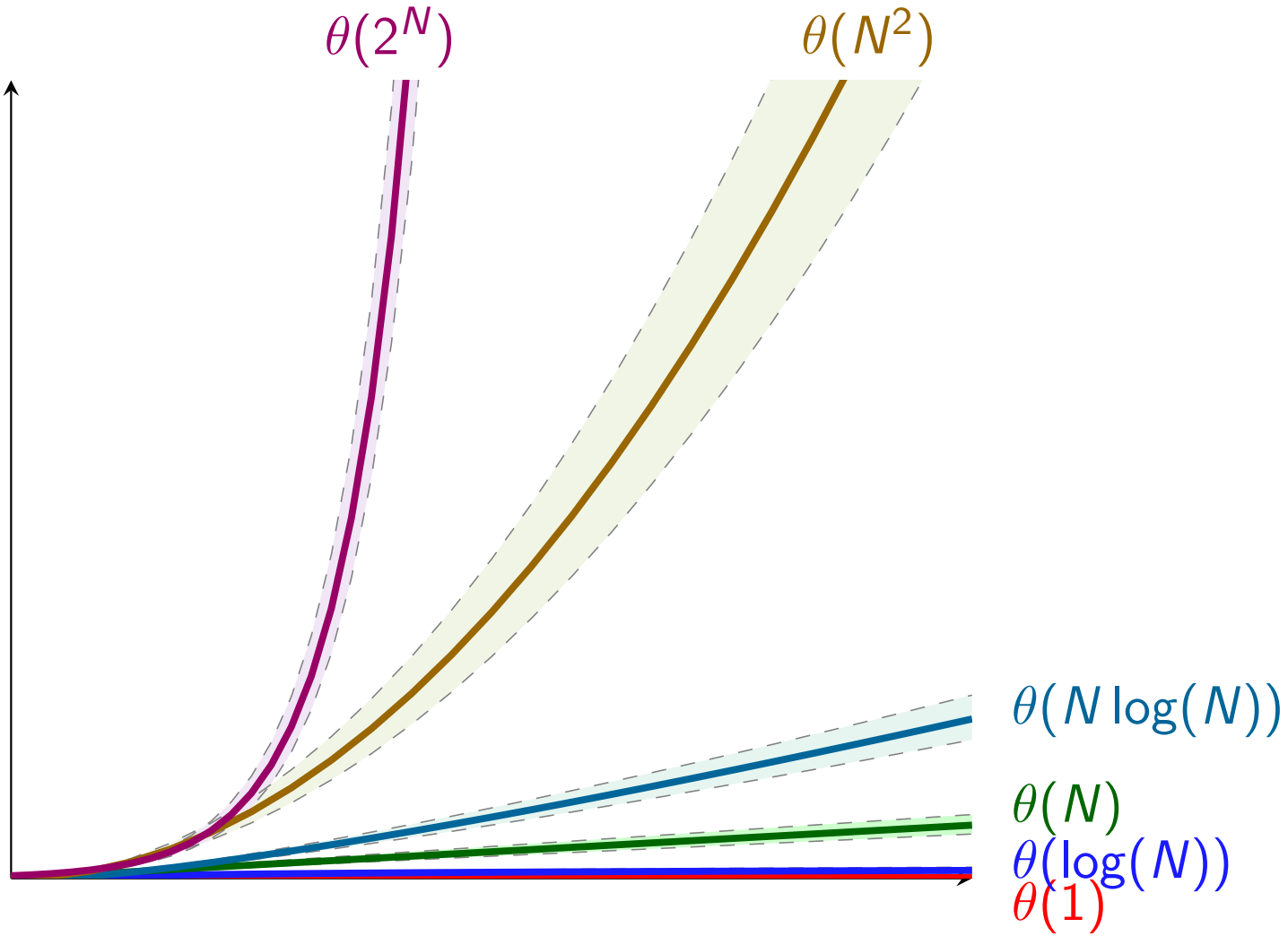
Reminders

- PA1 Tests due Sun, Sept 17 at 11:59 PM
  - Recitations will cover writing good test cases.
- PA1 Implementation due Sun, Sept 24 at 11:59 PM
  - Implement a Sorted Linked List

Complexity Classes



Zoom Out



Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Pick Algorithm 2 . . . usually.

1. Algorithm 1 is $\theta(N)$
2. Algorithm 2 is $\theta(N)$

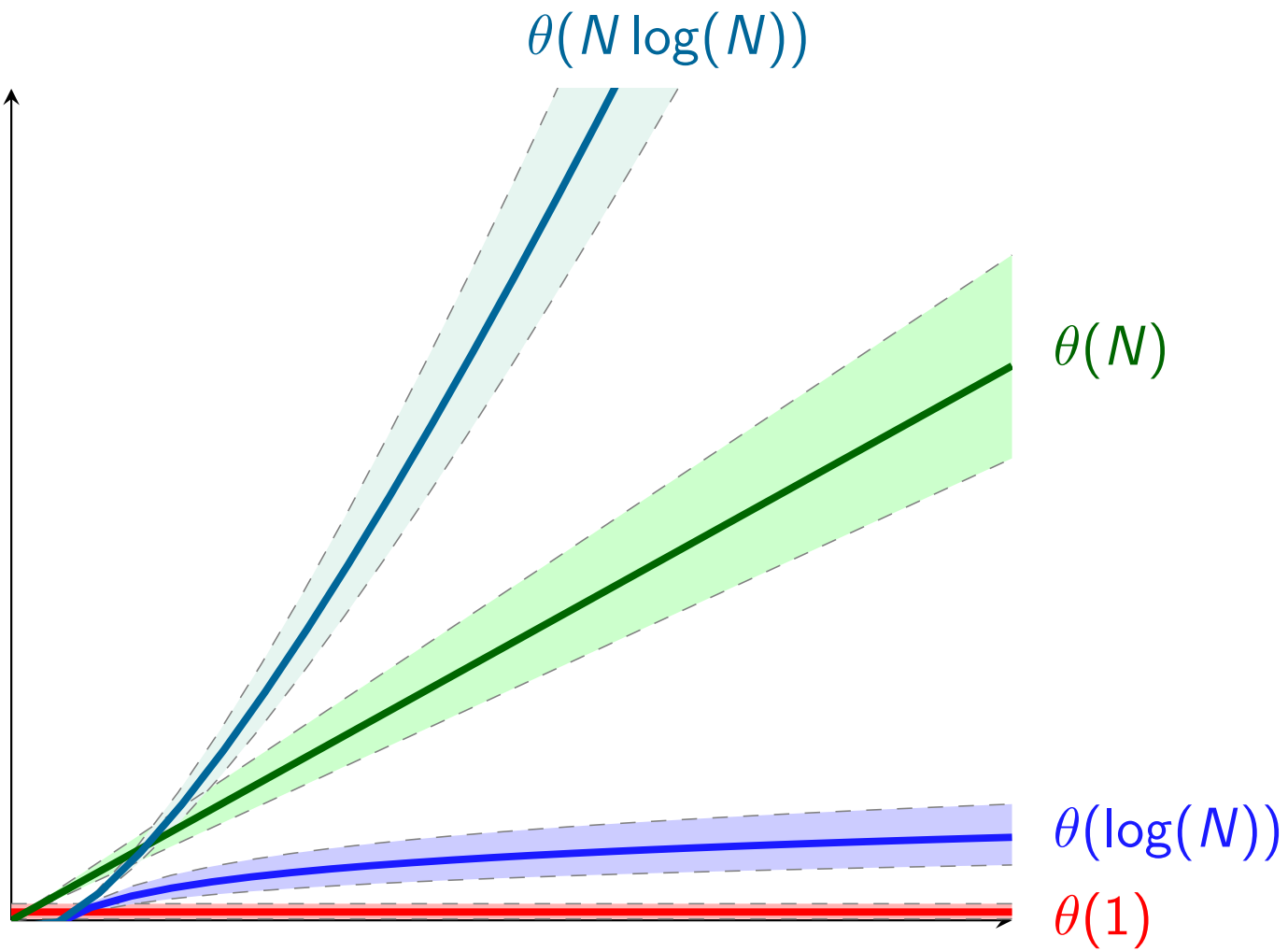
Measure actual runtimes

Examples

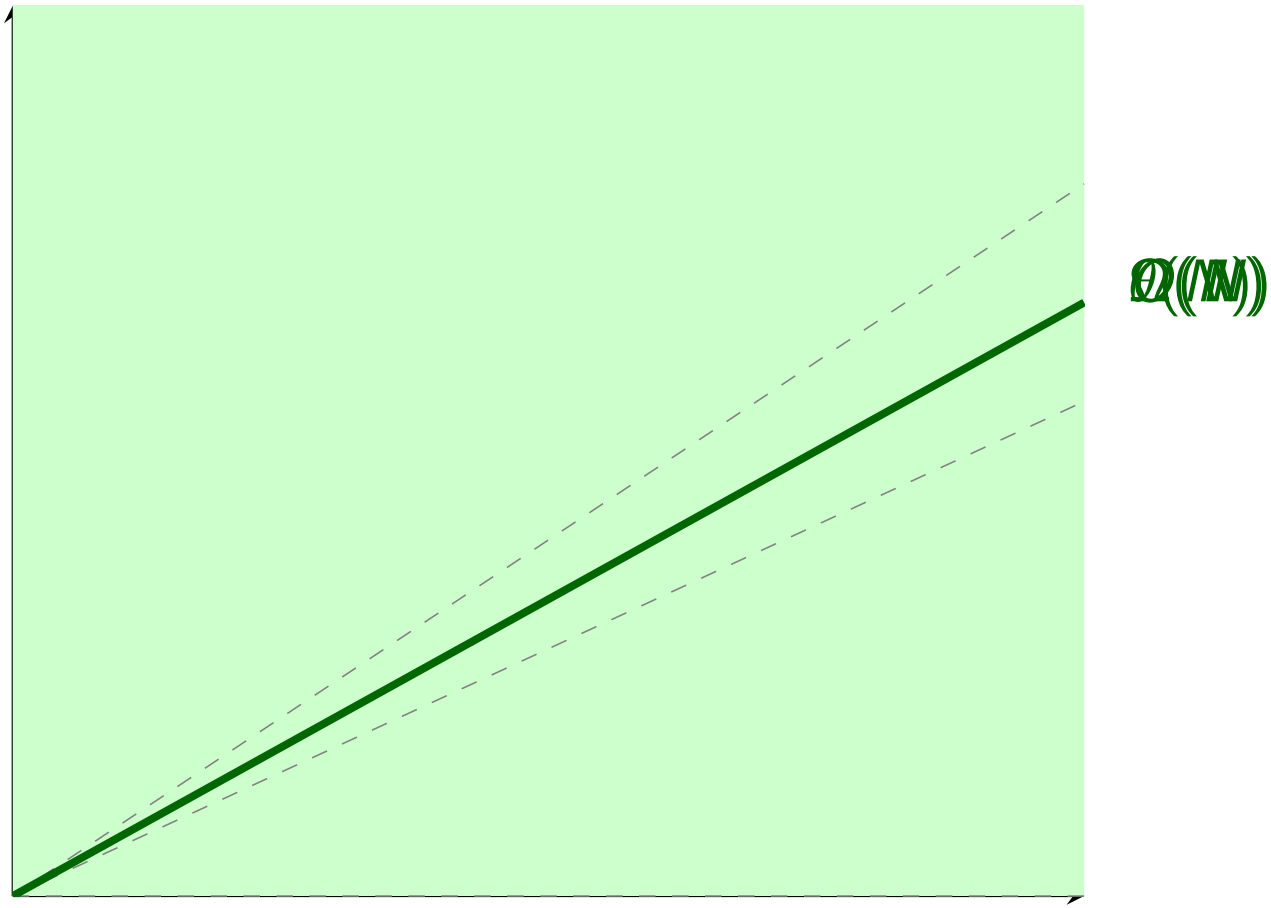
What is the asymptotic runtime of:

- Count the number of items in an N-item linked list.
  ($\theta(N)$)
- Count the number of times `x` appears in a linked list.
  ($\theta(N)$)
- Compute $x!$ ($= x \cdot (x-1) \cdot (x-2) \cdot \ldots \cdot 1$).
  ($\theta(x)$)

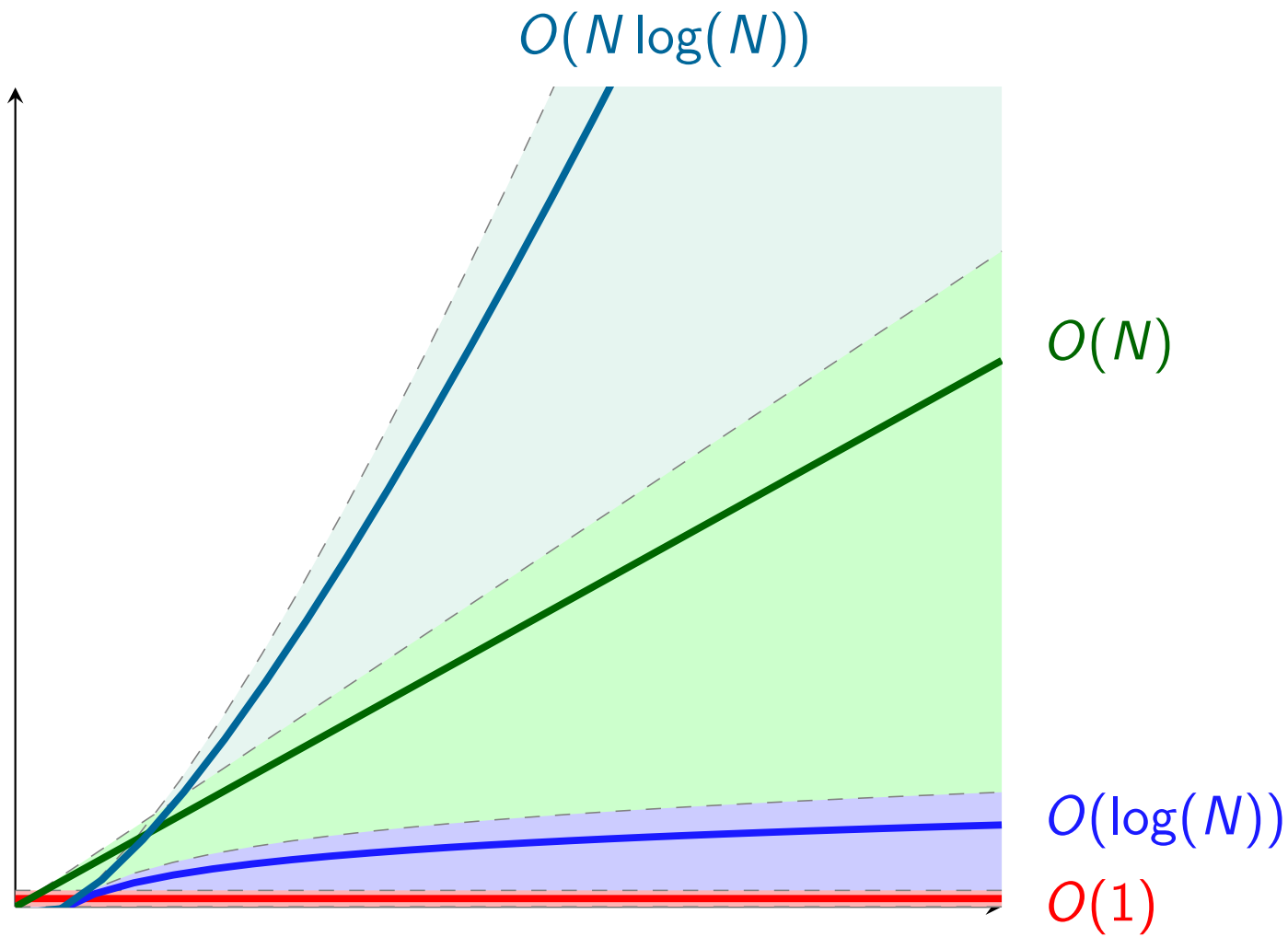
Complexity Bounds



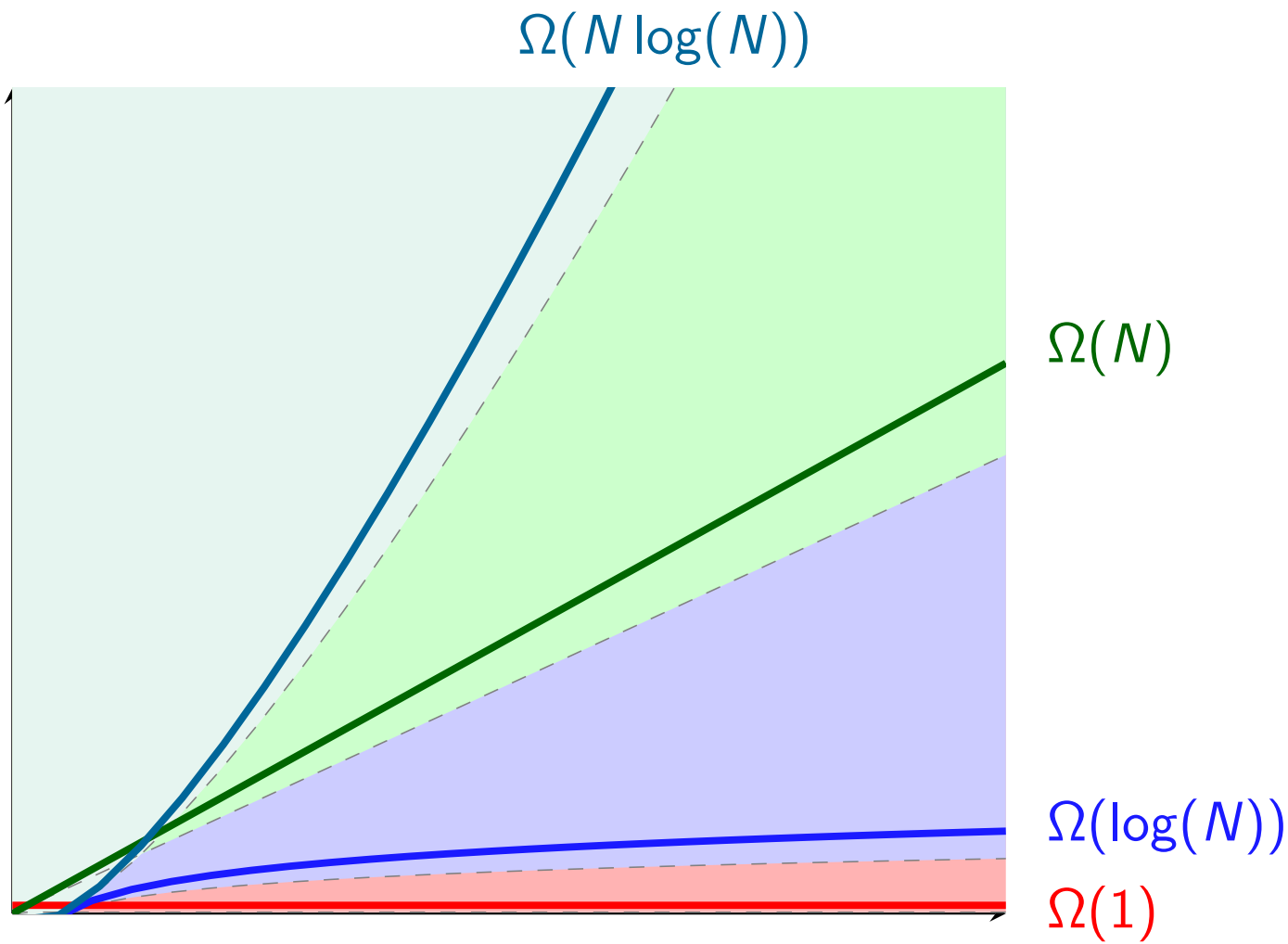
Complexity Bounds



Complexity Bounds



Complexity Bounds



Complexity Bounds

$g(N) \in O(f(N))$ if and only if:

- You can pick an N_0
- You can pick a c
- For all $N > N_0$: $g(N) \leq c \cdot f(N)$

$g(N) \in \Omega(f(N))$ if and only if:

- You can pick an N_0
- You can pick a c
- For all $N > N_0$: $g(N) \geq c \cdot f(N)$

Formalizing θ

When is $g(N) \in \theta(f(N))$?

Idea 1:
There exists some c_1 and c_2 that makes: $g(N) = c_1 + c_2 \cdot f(N)$

Problem: We want $N^2 + N \in \theta(N^2)$

Idea 2 (Use O, Ω)

$$g(N) \in O(f(N)) \textbf{ and } g(N) \in \Omega(f(N))$$

$$\updownarrow$$

$$g(N) \in \theta(f(N))$$

Examples

$$n^2 + 4n \stackrel{?}{\in} \theta(n^2)$$

$$2^n + 4n \stackrel{?}{\in} \theta(n^2)$$

$$1000 \cdot n\log(n) + 5n \stackrel{?}{\in} \theta(n\log(n))$$

Shortcut: Find the dominant term being summed, and compare it.

Tight Bounds

If $g(N) \in \theta(f(N))$:

- $g(N) \in O(f(N))$ is a **tight bound**.
- $g(N) \in \Omega(f(N))$ is a **tight bound**.

Examples

```
1  public void updateUsers(User[] users)
2  {
3    x = 1;
4    for(user : users)
5    {
6      user.id = x;
7    }
8  }
```

$$1 + \sum_{\texttt{user} \in \texttt{users}} 2 \text{ steps} = 1 + 2 \times |\texttt{users}| \in \theta(|users|N)$$

Examples

```
1  public void userFullName(User[] users, int id)
2  {
3    User user = users[id];
4    String fullName = user.firstName + user.lastName;
5    return fullName;
6  }
```

$$3 \in \theta(1)$$

Count the Steps

```
1   public void totalReads(User[] users, Post[] posts)
2   {
3     int totalReads = 0;
4     for(post : posts)
5     {
6       int userReads = 0;
7       for(user : users)
8       {
9         if(user.readPost(post)){ userReads += 1; }
10      }
11      totalReads += userReads;
12    }
13  }
```

$$1 + \sum_{\text{post} \in \text{posts}} \left(3 + \sum_{\text{user} \in \text{users}} 2 \right)$$

Count the Steps

$$1 + \sum_{\texttt{post} \in \texttt{posts}} \left(3 + \sum_{\texttt{user} \in \texttt{users}} 2 \right)$$

$$= 1 + \sum_{\texttt{post} \in \texttt{posts}} (3 + 2 \cdot |\texttt{users}|)$$

$$= 1 + \left(\sum_{\texttt{post} \in \texttt{posts}} 3 \right) + \left(\sum_{\texttt{post} \in \texttt{posts}} 2 \cdot |\texttt{users}| \right)$$

$$= 1 + (3 \cdot |\texttt{posts}|) + \left(\sum_{\texttt{post} \in \texttt{posts}} 2 \cdot |\texttt{users}| \right)$$

$$= 1 + (3 \cdot |\texttt{posts}|) + (2 \cdot |\texttt{users}| \cdot |\texttt{posts}|)$$

$$\in \theta(|\texttt{users}| \cdot |\texttt{posts}|)$$

Another Example

```
1   public int myAlgorithm(int[] input)
2   {
3     if(input.size % 2 == 0){
4       return 12345;
5     } else {
6       var total = 0;
7       for(i : input)
8       {
9         total += i;
10      }
11      return total;
12    }
13  }
```

Another Example

```
1   public int myAlgorithm(int[] input)
2   {
3     if(input.size % 2 == 0){
4       θ(1)
5     } else {
6       θ(1)
7       for(i : input)
8       {
9         θ(1)
10      }
11      θ(1)
12    }
13  }
```

Let's call $|input| = N$

Another Example

```
1   public int myAlgorithm(int[] input)
2   {
3     if(input.size % 2 == 0){
4       θ(1)
5     } else {
6       θ(1)
7       θ(N · 1)
8       θ(1)
9     }
10  }
```

Another Example

```
1    public int myAlgorithm(int[] input)
2    {
3      if(input.size % 2 == 0){
4        θ(1)
5      } else {
6        θ(N)
7      }
8    }
```

$\theta(1)$ if N is even **OR** $\theta(N)$ if N is odd.

Multi-Class Functions

$$T(N) = \begin{cases} \theta(1) & \textbf{if } N \text{ is even} \\ \theta(N) & \textbf{if } N \text{ is odd} \end{cases}$$

What is the complexity class of $T(N)$?

- $T(N) \in O(N)$ is a **tight** bound.
- $T(N) \in \Omega(1)$ is a **tight** bound.

If the tight Big-O and Big-Ω bounds are different, the function is not in ANY complexity class. (Big-Theta doesn't exist).

Does Big-Theta Exist?

$N + 2N^2$ belongs to one complexity class. $(\theta(N^2))$

$5N + 10N^2 + 2^N$ belongs to one complexity class $(\theta(2^N))$

$$\begin{cases} 2^N & \textbf{if } \texttt{rand()} > 0.5 \\ N & \textbf{otherwise} \end{cases}$$ does **not** belong to one complexity class.

- **Usually $\theta(f_1(N) + f_2(N) + \ldots)$ is based on the dominant term**
- **If you see cases (i.e., '{'), it's probably multi-class.**