Algorithm Analysis II: Iterative

CSE 373 Winter 2020

Instructor: Hannah C. Tang

Teaching Assistants:

Aaron Johnston	Ethan Knutson	Nathan Lipiarski
Amanda Park	Farrell Fileas	Sam Long
Anish Velagapudi	Howard Xiao	Yifan Bai
Brian Chan	Jade Watkins	Yuma Tou
Elena Spasova	Lea Quan	

Announcements

- I don't have add codes
  - CS Advisors closed registration on Monday; if you're a senior and don't have an add code yet … let us know ASAP
  - Otherwise, 373 is closed
- I got my first anonymous feedback!
  - Can you please clarify what is "too fast"?
- HW1 is due, HW2 is released
  - Late policy: ~5% first day, ~10% second day, 20% third day
  - Survey coming out soon
- Hint for HW2's confusingTest(): commenting out resize() is not the answer

Questions from Reading Quiz

- I'm still confused how N(N-1)/2 was derived
- It's not easy to find the exact count/cost, and it's especially easy to be off by one. Will we have to calculate the exact count at some point?
- I found it interesting how selection sort has no best or worst case
- Are there any better ways to sort? N^2 seems like a pretty bad runtime.

Derivation: N(N-1)/2

$$C = \overbrace{1 + 2 + 3 + \cdots + (N-3) + (N-2) + (N-1)}^{N-1 \text{ terms}}$$

$$+ \; C = (N-1) + (N-2) + (N-3) + \cdots + 3 + 2 + 1$$

$$2C = N + N + N + \cdots + N + N + N$$

$$2C = (N-1)\, N$$

$$\therefore \quad C = N(N-1)/2$$

Lecture Outline

- **Intro to Deques**
- Asymptotic Analysis Formalisms: O, Θ, and Ω
- Case Analyses: Best, Worst, Overall
- Asymptotic Analysis Case Study: PrintParty

Deques

Deque ADT. A collection storing an ordered sequence of elements.

- Each element is accessible by a zero-based index.
- A deque has a size defined as the number of elements in the deque.
- Elements can be added to the front or back.
- Optionally, elements can be removed from the front or back.

List ADT. A collection storing an ordered sequence of elements.

- Each element is accessible by a zero-based index.
- A list has a size defined as the number of elements in the list.
- Elements can be added to the front, back, *or any index in the list*.
- Optionally, elements can be removed from the front, back, *or any index in the list*.

Deques

- What series of List ADT methods or Deque ADT methods can build this sequence:
  - Deque has addFirst, addLast
  - List has those two, plus addAt(idx, val) and reserve(capacity)

A	B	C	D

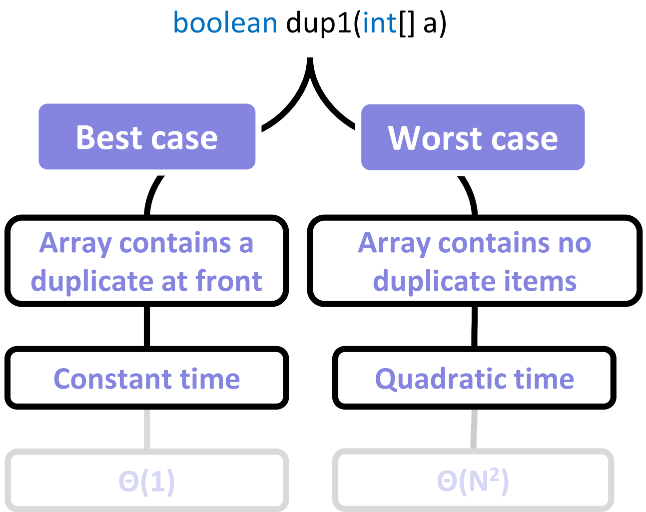
- Example:
  - addFirst(D), addFirst(C), addFirst(B), addFirst(A)
  - Works for either Deque or List

Lecture Outline

- Intro to Deques
- **Asymptotic Analysis Formalisms: Θ, O, and Ω**
- Case Analyses: Best, Worst, Overall
- Asymptotic Analysis Case Study: PrintParty

Review: Algorithm Analysis: Our Destination

- "The worst-case order-of-growth for dup1's runtime is quadratic (parabolic)"
- "The best-case order-of-growth for dup1's runtime is constant"



Big-O: Intuition

- Suppose we have a function R(N) with order of growth f(N). In Big-O notation, we write this as: "element of"

$$R(N) \boxed{\in} O(f(N))$$

- Big-O can informally be thought of as something like "less-than or equals"

Function	Big-O
$N^3 + 3N^4$	$O(N^4)$
$(1 / N) + N^3$	$O(N^3)$
$Ne^N + N$	$O(Ne^N)$
$40 \sin(N) + 4N^2$	$O(N^2)$

Big-O: Intuition

- … and also …

Function	Big-O	Also Big-O
$N^3 + 3N^4$ $\leq$	$O(N^4)$ $<$	$O(N^5)$
$(1 / N) + N^3$ $\leq$	$O(N^3)$ $<$	$O(N^{42342153154 2})$
$Ne^N + N$ $\leq$	$O(Ne^N)$ $<$	$O(N*3^N)$
$40 \sin(N) + 4N^2$ $\leq$	$O(N^2)$ $<$	$O(N^{2.1})$

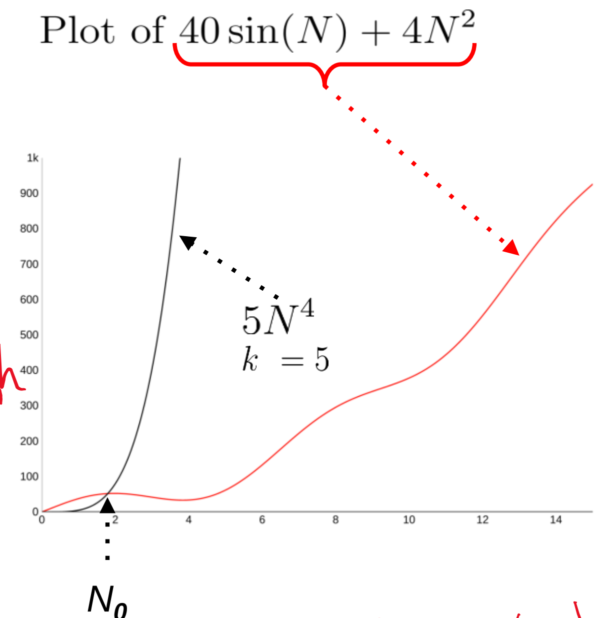
Big-O: Mathematical Definition

$$R(N) \in O(f(N))$$

means there exists a positive constant *k* such that

$$R(N) \leq k \cdot f(N)$$ "less than or equal to"

for all values of *N* greater than some N_0. "as we approach infinity"



$\therefore 40\sin(N) + 4N^2 \in O(N^4)$
(since $k=5$ and $N_0 \approx 2$)

Demo:
https://www.desmos.com/calculator/kaxnmhsjni

Big-Theta: Intuition

- Whereas Big-O can informally be thought of as something like "less-than or equals", Big-Theta more closely resembles "equals"

$$R(N) \in \Theta(f(N))$$

Function	Big-Theta
$N^3 + 3N^4$	$\Theta(N^4)$
$(1 / N) + N^3$	$\Theta(N^3)$
$Ne^N + N$	$\Theta(Ne^N)$
$40 \sin(N) + 4N^2$	$\Theta(N^2)$

Big-Theta: Intuition

- … but not …

Function	Big-Theta	Big-O (but not big-theta)
$N^3 + 3N^4$	$\Theta(N^4)$	$O(N^5)$
$(1 / N) + N^3$	$\Theta(N^3)$	$O(N^{423421531542})$
$Ne^N + N$	$\Theta(Ne^N)$	$O(N*3^N)$
$40 \sin(N) + 4N^2$	$\Theta(N^2)$	$O(N^{2.1})$

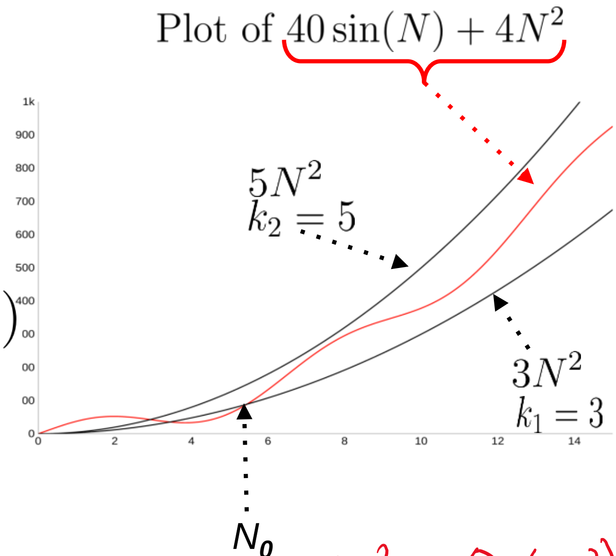
Big-Theta: Mathematical Definition

$$R(N) \in \Theta(f(N))$$

means there exist positive constants k_1 and k_2 such that

$$k_1 \cdot f(N) \le R(N) \le k_2 \cdot f(N)$$

for all values of N greater than some N_0.



Demo:
https://www.desmos.com/calculator/wsej7ymhtc

$\therefore$ $40\sin(N) + 4N^2 \in \Theta(N^2)$
since $k_1 = 3$, $k_2 = 5$, $N_0 \approx 5$

Big-Theta Challenge

Find a simple f(N) and corresponding k_1 and k_2.

$$R(N) = \frac{4N^2 + 3N \ln N}{2}$$

Remember: $R(N) \in \Theta(f(N))$

means there exist positive constants k_1 and k_2 such that

$$k_1 \cdot f(N) \leq R(N) \leq k_2 \cdot f(N)$$

for all values of N greater than some N_0.

Big-Omega: Intuition

- Big-Omega is our "greater than or equals"

$$R(N) \in \Omega(f(N))$$

Function	Big-Omega	Also Big-Omega
$N^3 + 3N^4$ $\geq$	$\Omega(N^4)$ $>$	$\Omega(N^3)$
$(1 / N) + N^3$ $\geq$	$\Omega(N^3)$ $>$	$\Omega(1)$
$Ne^N + N$ $\geq$	$\Omega(Ne^N)$ $>$	$\Omega(N*2^N)$
$40 \sin(N) + 4N^2$ $\geq$	$\Omega(N^2)$ $>$	$\Omega(N^{1.9})$

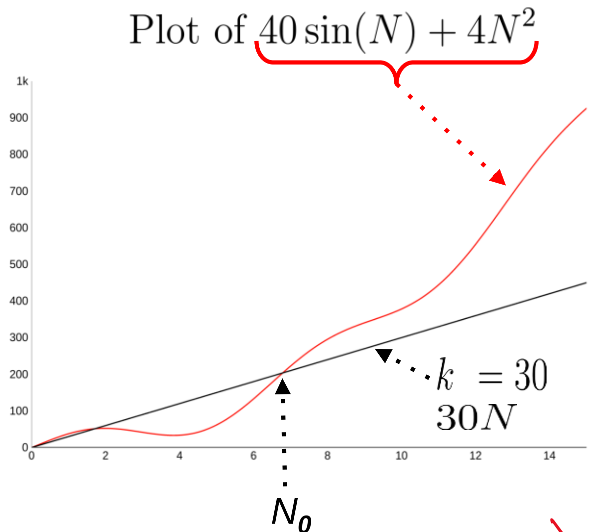
Big-Omega: Mathematical Definition

$$R(N) \in \Omega(f(N))$$

means there exists a positive constant *k* such that

$$k \cdot f(N) \leq R(N)$$

for all values of *N* greater than some N_0.



$\therefore 40\sin(N) + 4N^2 \in \Omega(N)$

since $k=30$ and $N_0 \approx 7$

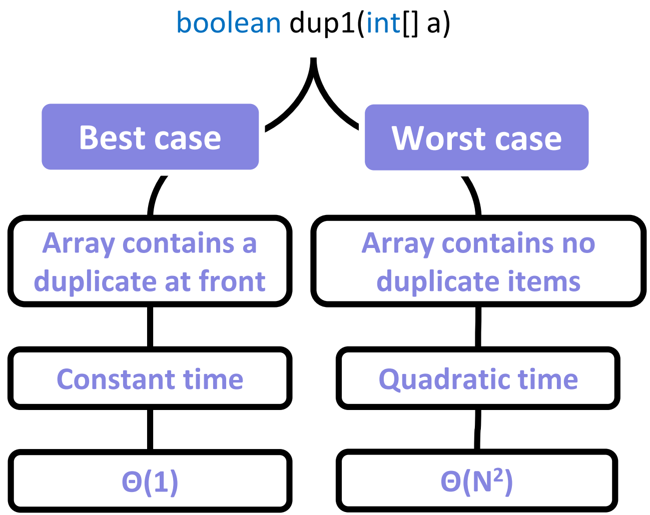
Big-O, Big-Theta, Big-Omega Relationship

- If a function f is in Big-Theta, what does it mean for its membership in Big-O and Big-Omega? Vice versa?

Function	Big-O	Big-Theta	Big-Omega
$N^3 + 3N^4$	$O(N^4)$	$\Theta(N^4)$	$\Omega(N^4)$
$(1 / N) + N^3$		$\Theta(N^3)$	
$Ne^N + N$		$\Theta(Ne^N)$	
$40 \sin(N) + 4N^2$		$\Theta(N^2)$	

🎉 Algorithm Analysis: Our Destination 🎉

- "The worst-case runtime for dup1 is $\Theta(N^2)$"
- "The best-case runtime for dup1 is $\Theta(1)$"
- Unless otherwise specified, we typically mean Big-Theta of worst case



Lecture Outline

- Intro to Deques
- Asymptotic Analysis Formalisms: O, Θ, and Ω
- **Case Analyses: Best, Worst, Overall**
- Asymptotic Analysis Case Study: PrintParty

Case Analysis, Redux

- Asymptotic analysis describes the function's behavior as it approaches infinity, without regard to its input
- Case analysis looks at a *specific input* or *class of inputs*

These analysis types are orthogonal!

- We've seen best case and worst case
- There is also the "all case" aka "overall case"
  - This is the case corresponding to *all* possible inputs
  - (there's also an amortized case, which we don't discuss in 373)

	O	Θ	Ω
best			
worst			
overall			

Overall Asymptotic Runtime Bound for dup1

→ all inputs, including best and worst

- Give overall asymptotic bounds for dup1's runtime
  - Reminder: $(N^2 + 3N + 2)/2$

Case	Big-O	Big-Theta	Big-Omega
Best		Θ(1)	
Worst		Θ(N²)	
Overall		∅	

Does not exist, since have different Θ-bounds for best and worst cases

- Demo: https://www.desmos.com/calculator/xjkrsyvxus

- Even though case analysis != asymptotic analysis, we can sometimes infer the overall case from asymptotic analyses

More Practice: Mystery

```
boolean mystery(int[] a, int target) {
    int N = a.length;
    for (int i = 0; i < N; i += 1)
        if (a[i] == target)
            return true;
    return false;
}
```

conditional early termination may indicate that an overall bound does not exist

Case	Big-Theta
Best	$\Theta(1)$
Worst	$\Theta(N)$
Overall	$\varnothing$

More Practice: Selection Sort

```
void selectionSort(int[] a) {
  int N = a.length;
  for (int i = 0; i < N; i++) {
    int smallestSoFar = a[i];
    for (int j = i + 1; j < N; j++) {
      if (a[j] < a[smallestSoFar]) {
        smallestSoFar = j;
      }
    }
    int tmp = a[i];
    a[i] = a[smallestSoFar];
    a[smallestSoFar] = tmp;
  }
}
```

Case	Big-Theta
Best	$\Theta(N^2)$
Worst	$\Theta(N^2)$
Overall	$\Theta(N^2)$

Lecture Outline

- Intro to Deques
- Asymptotic Analysis Formalisms: O, Θ, and Ω
- Case Analyses: Best, Worst, Overall
- **Asymptotic Analysis Case Study: PrintParty**

Poll Everywhere

pollev.com/uwcse373

Find an f(N) such that the runtime R(N) ∈ Θ(f(N)).

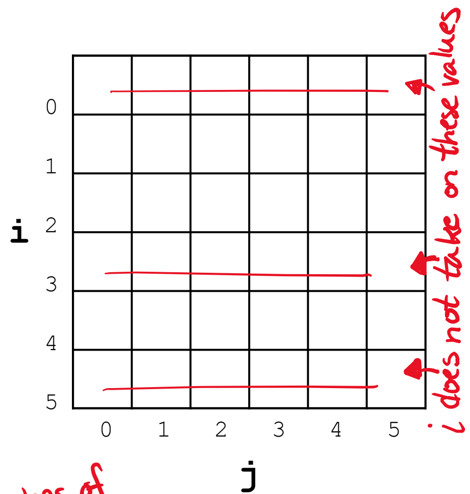
A. 1
B. log N
C. N
D. N log N
E. N^2
F. I'm not sure …

```java
void printParty(int N) {
  for (int i = 1; i <= N; i *= 2) {
    for (int j = 0; j < i; j += 1) {
      System.out.println("hello");
    }
  }
}
```

Why don't we have multiple cases to consider?

No conditional exits! So let's find the overall bound

PrintParty, Geometrically

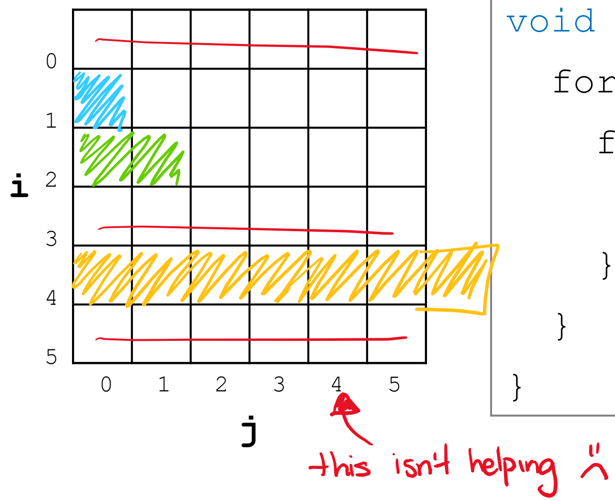



```java
void printParty(int N) {
   for (int i = 1; i <= N; i *= 2) {
      for (int j = 0; j < i; j += 1) {
         System.out.println("hello");
      }
   }
}
```

numbers of print stmts

N:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C(N):			same as		same as				same as		etc		etc	etc			same as	

PrintParty, Geometrically




```java
void printParty(int N) {
  for (int i = 1; i <= N; i *= 2) {
    for (int j = 0; j < i; j += 1) {
      System.out.println("hello");
    }
  }
}
```

N :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C(N) :	1	3	3	7	7	7	7	15	15	15	15	15	15	15	15	31	31	31

PrintParty, Counting

```
void printParty(int N) {
  for (int i = 1; i <= N; i *= 2) {
    for (int j = 0; j < i; j += 1) {
      System.out.println("hello");
    }
  }
}
```

$2^0=1$ $2^1=2$ $2^2=4$ $2^3=8$ $2^4=16$

N :	(1 + 2) + 3)			**4**	5	6	7 +	**8**)	9	10	11	12	13	14	15 +	**16**)	17	18
C(N) :	**1**	**3**	3	**7**	7	7	7	**15**	15	15	15	15	15	15	15	**31**	31	31 ...

floor of $\log_2 N$

$C(N) = 2^0 + 2^1 + 2^2 + \dots + 2^{\lfloor \log_2 N \rfloor}$

But what does this equal? :(

PrintParty, Using Examples

N	C(N)	½N	2N
1	1	0.5	2
4	1 + 2 + 4 = 7	2	14
7	1 + 2 + 4 = 7	3.5	14
8	1 + 2 + 4 + 8 = 15	4	16
27	1 + 2 + 4 + 8 + 16 = 31	13.5	54
185	... + 32 + 64 + 128 = 255	92.5	370
715	... + 256 + 512 = 1023	357.5	1430

let $k_1 = \frac{1}{2}$ and $k_2 = 2N$

$\therefore c(N) \in \Theta(N)$

tl;dr: Asymptotic Analysis for Iterative Problems

- Case Analysis != Asymptotic Analysis

- Memorize these summations since they're common:

$$1 + 2 + 3 + 4 + \ldots + (N-1) = N(N-1)/2 \in \Theta(N^2)$$

$$1 + 2 + 4 + 8 + \ldots + 2^{\mathrm{floor}(\log_2 N)} = 2N - 1 \in \Theta(N)$$

- Strategies for finding an asymptotic bound:
  - Use a geometric argument / visualizations
  - Find an expression for the exact step count
  - Write out examples