Intro to Algorithms

One-pass Algorithms

Sasha Golovnev

August 28, 2024

Teaching Assistants

Samuel King

Sidhant Saraogi

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106
- Next Class: Tuesday, 9/3

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106
- Next Class: Tuesday, 9/3
- Office Hours:
    - Tuesdays 11am, St. Mary's Hall 342C
    - Thursdays 2pm, St. Mary's Hall 354

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106
- Next Class: <u>Tuesday</u>, 9/3
- Office Hours:
  - Tuesdays 11am, St. Mary's Hall 342C
  - Thursdays 2pm, St. Mary's Hall 354
- `alexgolovnev+algo@gmail.com`

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106
- Next Class: Tuesday, 9/3
- Office Hours:
  - Tuesdays 11am, St. Mary's Hall 342C
  - Thursdays 2pm, St. Mary's Hall 354
- `alexgolovnev+algo@gmail.com`
- https://golovnev.org/algo/

Administrative Info

- Class Meetings: MW 12:30pm–1:45pm, ICC 106
- Next Class: Tuesday, 9/3
- Office Hours:
    - Tuesdays 11am, St. Mary's Hall 342C
    - Thursdays 2pm, St. Mary's Hall 354
- `alexgolovnev+algo@gmail.com`
- https://golovnev.org/algo/
- Canvas page for HW

Textbooks

- Primary Textbook:
  - *Cormen, Leiserson, Rivest, Stein.* Introduction to algorithms.

Textbooks

- Primary Textbook:
  - *Cormen, Leiserson, Rivest, Stein.* Introduction to algorithms.
- Additional textbooks:
  - *Dasgupta, Papadimitriou, Vazirani.* Algorithms.

Textbooks

- Primary Textbook:
  - *Cormen, Leiserson, Rivest, Stein.* Introduction to algorithms.
- Additional textbooks:
  - *Dasgupta, Papadimitriou, Vazirani.* Algorithms.
  - *Kulikov, Pevzner.* Ace Your Next Coding Interview by Learning Algorithms through Programming and Puzzle Solving.

Textbooks

- Primary Textbook:
  - *Cormen, Leiserson, Rivest, Stein.* Introduction to algorithms.
- Additional textbooks:
  - *Dasgupta, Papadimitriou, Vazirani.* Algorithms.
  - *Kulikov, Pevzner.* Ace Your Next Coding Interview by Learning Algorithms through Programming and Puzzle Solving.
  - *Kleinberg, Tardos.* Algorithm Design.

Textbooks

- Primary Textbook:
  - *Cormen, Leiserson, Rivest, Stein.* Introduction to algorithms.
- Additional textbooks:
  - *Dasgupta, Papadimitriou, Vazirani.* Algorithms.
  - *Kulikov, Pevzner.* Ace Your Next Coding Interview by Learning Algorithms through Programming and Puzzle Solving.
  - *Kleinberg, Tardos.* Algorithm Design.
  - *Erickson.* Algorithms. Online Draft.

Homework

- 5-6 Homework (50% of grade)

Homework

- 5-6 Homework (50% of grade)
- Midterm Exam (mid October) (25% of grade)

Homework

- 5-6 Homework (50% of grade)
- Midterm Exam (mid October) (25% of grade)
- Final Exam($\approx$ Dec 13) (25% of grade)

Homework

- 5-6 Homework (50% of grade)
- Midterm Exam (mid October) (25% of grade)
- Final Exam($\approx$ Dec 13) (25% of grade)
- *Extra Credit* for in-class Participation

Academic Integrity

- We take academic integrity seriously

Academic Integrity

- We take academic integrity seriously
- Permitted refs: listed textbook, lecture notes, course staff & students

Academic Integrity

- We take academic integrity seriously
- Permitted refs: listed textbook, lecture notes, course staff & students
- No search of solutions, No ChatGPT

Academic Integrity

- We take academic integrity seriously
- Permitted refs: listed textbook, lecture notes, course staff & students
- **No** search of solutions, **No** ChatGPT
- Can work in groups of 2-3 people, **must** write up your own solutions and list all group members

Topics Overview

- Divide and Conquer

Topics Overview

- Divide and Conquer
- Greedy Algorithms

Topics Overview

- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming

Topics Overview

- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Graph Algorithms

Topics Overview

- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Graph Algorithms
- Advanced Topics: Algorithms for Bid Data, Randomized and Approximation Algorithms

Today's Lecture

- Algorithmic Thinking

Today's Lecture

- Algorithmic Thinking
- Efficient algorithms

Today's Lecture

- Algorithmic Thinking
- Efficient algorithms
- Next Lecture: Asymptotic notation

Algorithms

- Pseudocode

Algorithms

- Pseudocode
- Correctness

Algorithms

- Pseudocode
- Correctness
- Running Time

Missing Number

Missing Number

- Input contains n distinct numbers in range $\{0, \ldots, n\}$

Missing Number

- Input contains n distinct numbers in range $\{0, \ldots, n\}$
- Return the only missing number

Missing Number

- Input contains n distinct numbers in range $\{0, \dots, n\}$
- Return the only missing number
- Efficient algorithm?

Streaming Algorithm

- Compute sum of all elements in stream:

$$s = x_1 + \ldots x_n$$

STREAMING ALGORITHM

- Compute sum of all elements in stream:

$$s = x_1 + \dots x_n$$

- Sum of all numbers in range $\{0, \dots, n\}$ is $S = \frac{n(n+1)}{2}$

Streaming Algorithm

- Compute sum of all elements in stream:

$$s = x_1 + \ldots x_n$$

- Sum of all numbers in range $\{0, \ldots, n\}$ is $S = \frac{n(n+1)}{2}$
- Missing number is $S - s = \frac{n(n+1)}{2} - s$

Streaming Algorithm

- Compute sum of all elements in stream:

$$s = x_1 + \ldots x_n$$

- Sum of all numbers in range $\{0, \ldots, n\}$ is $S = \frac{n(n+1)}{2}$
- Missing number is $S - s = \frac{n(n+1)}{2} - s$
- One pass through stream, efficient processing, $O(\log n)$ space

Two Missing Elements

- Stream contains $n - 1$ distinct numbers in range $\{0, \ldots, n\}$

Two Missing Elements

- Stream contains $n - 1$ distinct numbers in range $\{0, \ldots, n\}$
- Return both missing numbers

Two Missing Elements

- Stream contains $n - 1$ distinct numbers in range $\{0, \ldots, n\}$
- Return both missing numbers
- Efficient algorithm?

Streaming Algorithm

- Compute sum and sum of squares of all elements in stream:

$$s = x_1 + \dots x_{n-1}$$
$$t = x_1^2 + \dots x_{n-1}^2$$

Streaming Algorithm

- Compute sum and sum of squares of all elements in stream:

$$s = x_1 + \ldots x_{n-1}$$
$$t = x_1^2 + \ldots x_{n-1}^2$$

- Sum of all numbers in range $\{0, \ldots, n\}$ is $S = \frac{n(n+1)}{2}$
  Sum of squares of all numbers in range $\{0, \ldots, n\}$ is $T = \frac{n(n+1)(2n+1)}{6}$

Streaming Algorithm

- If missing numbers are a and b, then

$$a + b = S - s$$
$$a^2 + b^2 = T - t$$

Streaming Algorithm

- If missing numbers are a and b, then

$$\begin{aligned} a + b &= S - s \\ a^2 + b^2 &= T - t \end{aligned}$$

- One pass through stream, efficient processing, $O(\log n)$ space

Majority Element

Majority Element

- Stream has element occuring $> n/2$ times

Majority Element

- Stream has element occuring $> n/2$ times
- Find it!

Streaming Algorithm

- count $\leftarrow 0$; m $\leftarrow \perp$

Streaming Algorithm

- count $\leftarrow 0$;   m $\leftarrow \perp$
- For each element x_i of Stream:

Streaming Algorithm

- $\mathsf{count} \leftarrow 0$; $\mathsf{m} \leftarrow \perp$
- For each element x_i of Stream:
  - If $\mathsf{count} = 0$, then $\mathsf{m} \leftarrow x_i$ and $\mathsf{count} \leftarrow 1$

Streaming Algorithm

- $\mathsf{count} \leftarrow 0; \quad \mathsf{m} \leftarrow \perp$
- For each element x_i of Stream:
  - If $\mathsf{count} = 0$, then $\mathsf{m} \leftarrow x_i$ and $\mathsf{count} \leftarrow 1$
  - ElseIf $x_i = \mathsf{m}$, then $\mathsf{count}\,\text{++}$

Streaming Algorithm

- $\mathsf{count} \leftarrow 0$; $\mathsf{m} \leftarrow \perp$
- For each element x_i of Stream:
  - If $\mathsf{count} = 0$, then $\mathsf{m} \leftarrow x_i$ and $\mathsf{count} \leftarrow 1$
  - ElseIf $x_i = \mathsf{m}$, then count ++
  - Else count --

Streaming Algorithm

- $\mathsf{count} \leftarrow 0$; $\mathsf{m} \leftarrow \perp$
- For each element x_i of Stream:
  - If $\mathsf{count} = 0$, then $\mathsf{m} \leftarrow x_i$ and $\mathsf{count} \leftarrow 1$
  - ElseIf $x_i = \mathsf{m}$, then count ++
  - Else count --
- Return m

Example

Another View

Misra-Gries Algorithm

Misra-Gries Algorithm

- $\mathsf{count}_1, \ldots, \mathsf{count}_k \leftarrow 0; \quad \mathsf{m}_1, \ldots, \mathsf{m}_k \leftarrow \perp$
- For each element x_i of Stream:
  - If $x_i = \mathsf{m}_j$, then count_j ++
  - Else
    - Let count_j be min in $\mathsf{count}_1, \ldots \mathsf{count}_k$
    - If $\mathsf{count}_j = 0$, then $\mathsf{m}_j = x_i; \quad \mathsf{count}_j = 1$
    - Else count_1 --, …, count_k --
- Return $\mathsf{m}_1, \ldots, \mathsf{m}_k$