Sometimes Randomness is faster

Mohtassim Masud [2105153] Ahnaf Akib Tanim [2105154] Md.Shahjalal Rumman [2105165]

Department of CSE Bangladesh University of Engineering and Technology

December 3, 2024

### Introduction

#### Real life Scenario

Definition
Deterministic Algo.
CHaracteristics
RandomiZed Algo.
Characteristics

## Randomized Min Cut

Intro to Min Cut
Naive Algorithm

Classic Quick Sort and its optimization

Conclusion

# Before diving deep into Randomized Algorithms...

### Introduction

### Real life Scenario

Deterministic Algo. CHaracteristics RandomiZed Algo.

## Randomized Mi

Intro to Min Cut
Naive Algorithm
Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

## Before diving deep into Randomized Algorithms...

## Real-Life Applications of Randomness

How does NASA make decisions for Mars Rovers?

### Introduction

### Real life Scenario

Deterministic Algo.

RandomiZed Algo.

Characteristics

### Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

## Before diving deep into Randomized Algorithms...

## Real-Life Applications of Randomness

- How does NASA make decisions for Mars Rovers?
- How does Google secure cryptocurrency transactions?

### Introduction

### Real life Scenario

Definition
Deterministic Algo.

RandomiZed Algo.

### Randomized Mir

Intro to Min Cut Naive Algorithm Randomized Min Cut

### Classic Quick Sort and its optimization

Conclusion

## Before diving deep into Randomized Algorithms...

### Real-Life Applications of Randomness

- How does NASA make decisions for Mars Rovers?
- How does Google secure cryptocurrency transactions?





### These examples reveal the power of randomness!

# Notation

## Introduction Real life Scenario

\_ \_

#### Definition

Deterministic Algo. CHaracteristics RandomiZed Algo. Characteristics

## Randomized Min

Intro to Min Cut Naive Algorithm

Randomized Min Cut

### Classic Quick Sort and its optimization

Conclusion

## Definition (Randomized Algorithms)

A randomized algorithm is an algorithm that incorporates randomness as part of its operation.

### Introduction

#### Deterministic Algo. CHaracteristics

RandomiZed Algo.

#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness

### Introduction

#### Deterministic Algo. CHaracteristics

RandomiZed Algo.

### Randomized Min

### Cut Intro to Min Cut

Naive Algorithm

Classic Quick Sort and its

optimization

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness
Objective	Exact solutions with good worst-case behavior

#### Introduction

Deterministic Algo. CHaracteristics

RandomiZed Algo.

#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness
Objective	Exact solutions with good worst-case behavior
Complexity	Often more complex with nuanced correctness proofs

#### Introduction

Real life Scenario

#### Deterministic Algo. CHaracteristics

RandomiZed Algo.

### Randomized Min

### Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness
Objective	Exact solutions with good worst-case behavior
Complexity	Often more complex with nuanced correctness proofs
Performance	Good in worst-case scenarios

### Introduction

Real life Scenario

#### Deterministic Algo. CHaracteristics

RandomiZed Algo.

### Randomized Min

### Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness
Objective	Exact solutions with good worst-case behavior
Complexity	Often more complex with nuanced correctness proofs
Performance	Good in worst-case scenarios

#### Introduction

Deterministic Algo. CHaracteristics

RandomiZed Algo.

Randomized Min

### Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Conclusion

Aspect	Deterministic Algorithm
Definition	Predefined set of rules, no randomness
Objective	Exact solutions with good worst-case behavior
Complexity	Often more complex with nuanced correctness proofs
Performance	Good in worst-case scenarios
Error Tolerance	Guaranteed correctness

Table: Characteristics of Deterministic Algorithms

### Introduction

Definition

Deterministic Algo.

RandomiZed Algo. Characteristics

## Randomized Min

Intro to Min Cut Naive Algorithm

Randomized Min Cu
Classic Quick

Sort and its optimization

Conclusion

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation

Table: Characteristics of Randomized Algorithms

Mohtassim Masud Ahnaf Akib Tanim Md.Shahjalal Rumman Randomised AlgorithmsDecember 3, 2024 5/4

### Introduction

Real life Scenario

Deterministic Algo.

RandomiZed Algo.

Characteristics

#### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation
Objective	High probability of close-to-correct solutions

### Introduction

Definition

Deterministic Algo.

RandomiZed Algo. Characteristics

## Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation
Objective	High probability of close-to-correct solutions
Complexity	Simpler and denser analyses

Table: Characteristics of Randomized Algorithms

Mohtassim Masud Ahnaf Akib Tanim Md.Shahjalal Rumman Randomised AlgorithmsDecember 3, 2024

### Introduction

Deterministic Algo.

RandomiZed Algo.

### Characteristics

### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation
Objective	High probability of close-to-correct solutions
Complexity	Simpler and denser analyses
Performance	Optimized for good average-case performance

### Introduction

Deterministic Algo.

RandomiZed Algo. Characteristics

#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation
Objective	High probability of close-to-correct solutions
Complexity	Simpler and denser analyses
Performance	Optimized for good average-case performance

### Introduction

Deterministic Algo.

RandomiZed Algo. Characteristics

### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Classic Quick Sort and its optimization

Conclusion

Aspect	Randomized Algorithm
Definition	Incorporates randomness in its operation
Objective	High probability of close-to-correct solutions
Complexity	Simpler and denser analyses
Performance	Optimized for good average-case performance
Error Tolerance	Small probability of error, adjustable by repetitions

Table: Characteristics of Randomized Algorithms

# What is Min-Cut?

### Introduction

Real life Scenario

Deterministic Algo.

RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

### Goal

Divide graph G = (V, E) into two parts. Minimize edges between parts.

### Uses

- Break networks.
- Cluster data.
- Optimize circuits.

### Introduction

Definition

Deterministic Algo. CHaracteristics

RandomiZed Algo.

### Characteristics

# Randomized Min Cut

Intro to Min Cut
Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

## **Steps**

Randomly pick an edge.

## Introduction Real life Scenario

Definition

Deterministic Algo.

RandomiZed Algo.

Characteristics

# Randomized Min Cut

Intro to Min Cut
Naive Algorithm

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

## Steps

- Randomly pick an edge.
- Merge the two vertices connected by the chosen edge.

#### Introduction Real life Scenario

Deterministic Algo.

RandomiZed Algo.

### Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut Classic Quick

### Sort and its optimization

Conclusion

## Steps

- Randomly pick an edge.
- Merge the two vertices connected by the chosen edge.
- Repeat the process until only two vertices remain.

## Introduction Real life Scenario

Definition

Definition

Deterministic Algo.

CHaracteristics

RandomiZed Algo. Characteristics

### Randomized Mi

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

### Classic Quick Sort and its optimization

Conclusion

## Steps

- Randomly pick an edge.
- Merge the two vertices connected by the chosen edge.
- Repeat the process until only two vertices remain.
- Perform this procedure for all possible edge combinations to find the minimum cut.

### Introduction

Real life Scenario Deterministic Algo.

RandomiZed Algo.

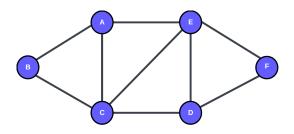
### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

### Conclusion



### Introduction

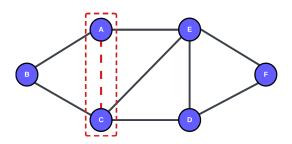
Deterministic Algo. RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



### Introduction

Deterministic Algo.

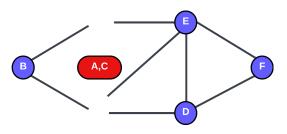
RandomiZed Algo.

#### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its

optimization Conclusion



### Introduction

Deterministic Algo.

RandomiZed Algo.

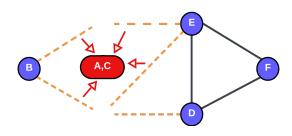
### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

### Classic Quick Sort and its optimization

Conclusion



### Introduction

Deterministic Algo.

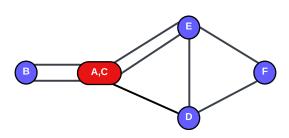
RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

### Conclusion



### Introduction

Deterministic Algo.

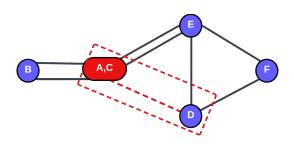
RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Order of Selection: AC,AC-D

### Introduction

Deterministic Algo.

RandomiZed Algo.

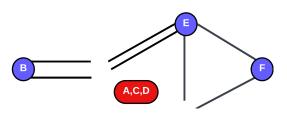
### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

### Conclusion



### Introduction

Deterministic Algo.

RandomiZed Algo.

### Randomized Min

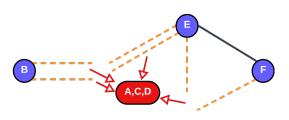
Intro to Min Cut

Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

#### Conclusion



### Introduction

Deterministic Algo.

RandomiZed Algo.

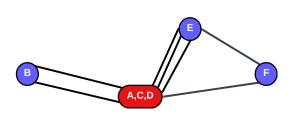
### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

### Conclusion



### Introduction

Definition

Deterministic Algo. CHaracteristics

RandomiZed Algo. Characteristics

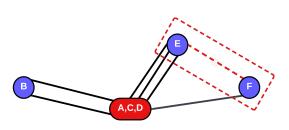
# Randomized Min Cut

Intro to Min Cut
Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Order of Selection: AC,ACD,E-F

### Introduction

Deterministic Algo.

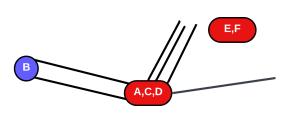
RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Order of Selection: AC,ACD,EF

### Introduction

Deterministic Algo.

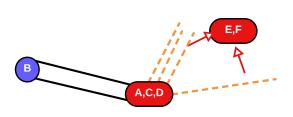
RandomiZed Algo.

### Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Order of Selection: AC,ACD,EF

### Introduction

Definition

Deterministic Algo.

RandomiZed Algo.

# Randomized Min Cut

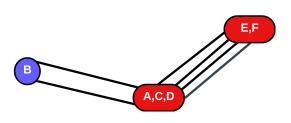
Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

### Conclusion



Order of Selection: AC,ACD,EF

#### Introduction

Real life Scenario

Definition

Deterministic Algo.

RandomiZed Algo.

## Randomized Min Cut

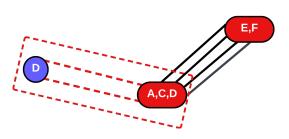
Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Order of Selection: AC,ACD,EF,ACD-B

#### Introduction

Real life Scenario Definition

Deterministic Algo.

RandomiZed Algo.

## Randomized Min Cut

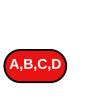
Intro to Min Cut

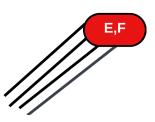
Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion





Order of Selection: AC,ACD,EF,ABCD

#### Introduction

Real life Scenario Definition

Definition

Deterministic Algo.

CHaracteristics

RandomiZed Algo. Characteristics

## Randomized Min Cut

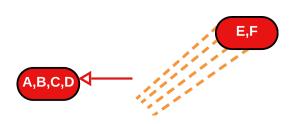
Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

#### Conclusion



Order of Selection: AC,ACD,EF,ABCD-EF

#### Introduction

Deterministic Algo.

RandomiZed Algo.

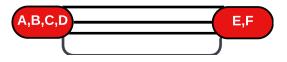
#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Min-Cut: ABCD and EF

# Naive Algorithm

### Introduction

Deterministic Algo.

RandomiZed Algo.

#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its

optimization Conclusion

# **Problems**

Mohtassim Masud Ahnaf Akib Tanim Md.Shahjalal Rumman Randomised AlgorithmsDecember 3, 2024

# Naive Algorithm

### Real life Scenario

Definition
Deterministic Algo.

CHaracteristics RandomiZed Algo. Characteristics

#### Randomized Mi Cut

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

### **Problems**

- Highly time-consuming.
- Requires exploring all possible combinations.
- Time complexity:  $\mathcal{O}(n^2 \cdot 2^m)$ , where n is the number of vertices and m is the number of edges.

# Naive Algorithm

### Real life Scenario

Definition

Deterministic Algo. CHaracteristics

RandomiZed Algo. Characteristics

### Cut

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion

### **Problems**

- Highly time-consuming.
- Requires exploring all possible combinations.
- Time complexity:  $\mathcal{O}(n^2 \cdot 2^m)$ , where n is the number of vertices and m is the number of edges.

### Next?

Let's improve it using \*\*randomized cuts\*\* for better efficiency.

## Towards Randomized Min-Cut

### Real life Scenario

Definition

Deterministic Algo.

CHaracteristics

RandomiZed Algo.

### Randomized M

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

## Classic Quick

Sort and its optimization

Conclusion

### Key Idea: Randomized Algorithm

### Randomness can help:

- Explore diverse contraction orders.
- Increase the chance of finding a true min-cut.

A Monte Carlo algorithm guarantees high success probability.

#### Introduction

Real life Scenario

Definition

Deterministic Algo.

RandomiZed Algo.

Characteristics

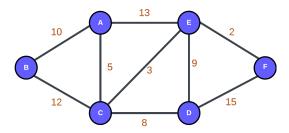
## Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



#### Introduction

Real life Sce

Deficience

Deterministic Algo.

CHaracteristics

RandomiZed Algo. Characteristics

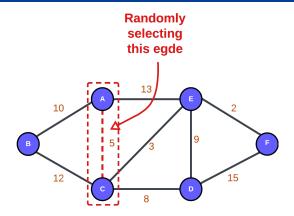
### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



### Selected Edge: AC

#### Introduction

Real life Scenario

Deterministic Algo.

RandomiZed Algo.

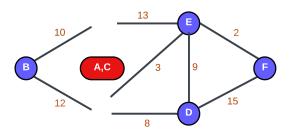
#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

#### Conclusion



### Merge A,C

#### Introduction

Deterministic Algo.

RandomiZed Algo.

#### Randomized Min Cut

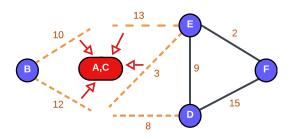
Intro to Min Cut

Naive Algorithm Randomized Min Cut

### Classic Quick Sort and its

optimization

#### Conclusion



Merge Adjacent edges of A,C

#### Introduction

Deterministic Algo.

RandomiZed Algo.

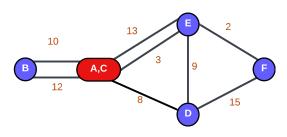
#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



### Merged Adjacent edges of A,C

#### Introduction

Real life Scenario

Definition

Deterministic Algo.

RandomiZed Algo.

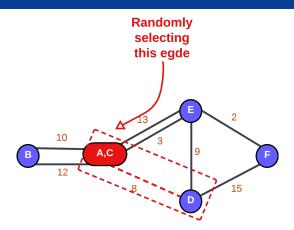
### Randomized Min

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



### Select edge A,C,D

### Introduction

Deterministic Algo.

RandomiZed Algo.

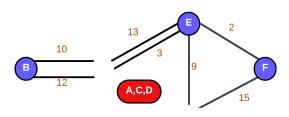
#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merge the vertices A,C,D

#### Introduction

Deterministic Algo.

RandomiZed Algo.

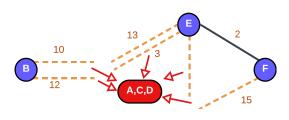
#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merge the adjacent edges of A,C,D

### Introduction

Definition

Deterministic Algo.

RandomiZed Algo.

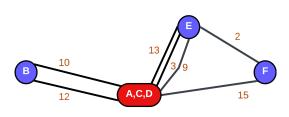
## Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



### Merged the edges

#### Introduction

Deterministic Algo.

RandomiZed Algo.

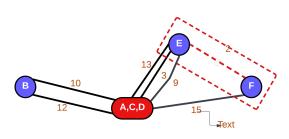
#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Select the edge E,F

### Introduction

Definition

Deterministic Algo. CHaracteristics

RandomiZed Algo.

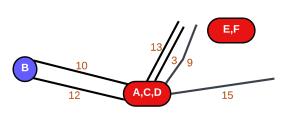
## Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merge the vertices E,F

#### Introduction

Deterministic Algo.

RandomiZed Algo.

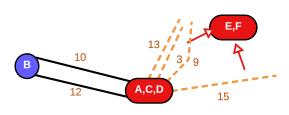
#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merge the Adjacent edges

### Introduction

Definition

Deterministic Algo.

RandomiZed Algo.

RandomiZed Algo Characteristics

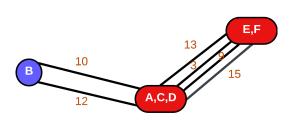
## Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merged the adjacent edges

#### Introduction

Deterministic Algo.

RandomiZed Algo.

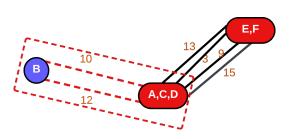
#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

### Classic Quick Sort and its

optimization Conclusion



Select Edge B, ACD

### Introduction

Deterministic Algo.

RandomiZed Algo.

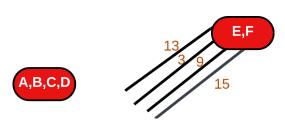
#### Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Merge the vertices B,ACD

### Introduction

Deterministic Algo.

RandomiZed Algo.

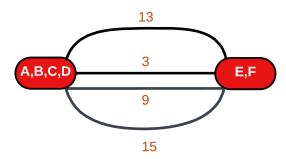
#### Randomized Min Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

Classic Quick Sort and its optimization

Conclusion



Only Two Vertices Remain;

## Choose the Minimum

Randomized Algorithms

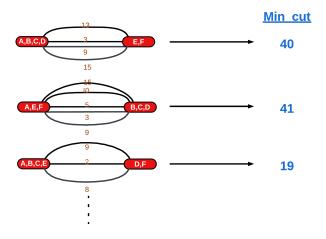
### Randomized Min

Cut

Intro to Min Cut

Randomized Min Cut

### and its



## Time Complexity: Min-Cut Algorithm

#### Introduction

Real life Scenario Definition

Deterministic Algo.

CHaracteristics RandomiZed Algo.

### Randomized Min

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

#### Classic Quick Sort and its optimization

Conclusion

- Efficient Algorithm for finding Minimum Cut.
- Time Complexity:  $O(E \cdot \log(V)^2)$ 
  - E: Edges in the graph.
  - *V*: **Vertices** in the graph.

## Summary

#### Introduction

Real life Example of Randomness

Randomized Algorithms

### Randomized Min

Cut Intro to Min Cut

Naive Algorithm

Randomized Min Cut

## Classic Quicksor

### Optimization

Quicksort

Time Complexity

- **■** Explored Min-Cut Algorithm.
- Key Aspects:

## Summary

#### Introduction

Real life Example of Randomness

Deterministic Algorithms Randomized Algorithms

### Randomized Min

Intro to Min Cut

Naive Algorithm

#### Randomized Min Cut

## Classic Quicksort and its

Outobsort

The Algorithm
Time Complexity

**■** Explored Min-Cut Algorithm.

- Key Aspects:
  - Divide graph into disjoint subsets.
  - Use flow-based methods.

## **Summary**

#### Introductio

Randomness
Deterministic Algorithm

Randomized Algorithms

### Randomized Min

Intro to Min Cut Naive Algorithm

#### Randomized Min Cut

## Classic Quicksort

Optimization

The Algorithm
Time Complexity

- Explored Min-Cut Algorithm.
- Key Aspects:
  - Divide graph into disjoint subsets.
  - Use flow-based methods.
- **Efficient:**  $O(E \cdot \log(V)^2)$  for large, sparse graphs.
- Conclusion: Practical, impactful for graph problems.

#### Introduction

Real life Example of Randomness Deterministic Algorithms

### Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

## and its Optimization

#### Quicksort

The Algorithm Time Complexity Optimization

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case: O(n log n) (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

### **Key Insight**

#### Introductio

Real life Example of Randomness

Deterministic Algorithms Randomized Algorithms

### Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quicksor and its

### Ouicksort

The Algorithm Time Complexity

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

### **Key Insight**

#### Introductio

Real life Example of Randomness

Deterministic Algorithms Randomized Algorithms

### Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quicksor and its

### Ouicksort

The Algorithm Time Complexity

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

**Key Insight** 

#### Introductio

Real life Example of Randomness

Randomized Algorithms

### Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

Classic Quicksor and its

### Ouicksort

The Algorithm
Time Complexity

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

**Key Insight** 

#### Introductio

Real life Example of Randomness

Randomized Min

Intro to Min Cut
Naive Algorithm
Randomized Min Cut

Classic Quicksor and its

### Optimization Ouicksort

The Algorithm
Time Complexity

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

### **Key Insight**

#### Introduction

Real life Example of Randomness

Randomized Algorithms

## Randomized Min Cut

Intro to Min Cut Naive Algorithm Randomized Min Cut

and its

#### Optimization Ouicksort

The Algorithm
Time Complexity
Ontimization

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

### **Key Insight**

#### Introduction

Real life Example of Randomness

Randomized Algorithms

### Randomized Min Cut

Naive Algorithm Randomized Min Cut

and its
Optimization

#### Quicksort

The Algorithm
Time Complexity
Optimization

### **Divide and Conquer:**

- Recursive partitions
- Independent sorts

### **Time Complexity:**

- Best/Average Case:  $O(n \log n)$  (Balanced splits)
- Worst Case:  $O(n^2)$  (Highly unbalanced splits)

### **Key Insight**

## **QuickSort Algorithm**

### Introductio

Real life Example of

Deterministic Algorithm

### Randomized Min

Intro to Min Cut Naive Algorithm Randomized Min Cut

Randomized Min Cut

### and its

### Quicksort

The Algorithm
Time Complexit

# Algorithm QuickSort (Divide and Conquer)

- 1: **Input:** Array A, indices low, high
- 2: Output: Sorted Array
- 3: **if** low < high **then**
- 4: pivot  $\leftarrow$  Partition(A, low, high)
- 5: QuickSort(A, low, pivot 1)
- 6: QuickSort(A, pivot + 1, high)
- 7: **end if**

### Algorithm Partition (Pivot Selection)

- 1: **Input:** Array A, indices
- 2: Output: Pivot index
- 3:  $pivot \leftarrow A[high]$
- $4: i \leftarrow low 1$
- 5: **for** i = low **to** high 1 **do** 
  - 6: **if**  $A[j] \leq pivot$  **then**
- 7:  $i \leftarrow i + 1$
- 8: Swap A[i], A[j]
- 9: end if
- 10: end for
- 11: Swap A[i + 1], A[high]
- 12: **return** *i* +

## **QuickSort Algorithm**

### Introduction

Real life Example of

Deterministic Algorithm

### Randomized Min Cut

Naive Algorithm Randomized Min Cut

Classic Ouicksor

### and its Optimization

### Quicksort

The Algorithm
Time Complexity
Optimization

# Algorithm QuickSort (Divide and Conquer)

- 1: **Input:** Array A, indices low, high
- 2: Output: Sorted Array
- 3: **if** low < high **then**
- 4: pivot ← Partition(A, low, high)
- 5: QuickSort(A, low, pivot 1)
- 6: QuickSort(A, pivot + 1, high)
- 7: end if

# Algorithm Partition (Pivot Selection)

- 1: **Input:** Array A, indices low, high
- 2: Output: Pivot index
- 3:  $pivot \leftarrow A[high]$
- 4:  $i \leftarrow low 1$
- 5: for j = low to high 1 do
- 6: **if**  $A[j] \leq pivot$  **then**
- 7:  $i \leftarrow i + 1$
- 8: Swap A[i], A[j]
- 9: end if
- 10: **end for**
- 11: Swap A[i + 1], A[high]
- 12: **return** *i* + 1

Randomized Min

Intro to Min Cut

Randomized Min Cut

### and its

The Algorithm

Pivot: ☐ i: ☐ j: ☐ Swap: ☐

i=-1

5

4

```
Introduction
```

Randomness

candomness

Randomized Algorithm

### Randomized Min

Intro to Min Cut

Intro to Min Cu

Randomized Min Cut

## Classic Quicksort and its

### Optimization

The Alexandre

The Algorithm

Ontimization

```
pivot \leftarrow A[high]
 \leftarrowlow – 1
for j = low to high -1 do
   if A[j] < pivot then
         i \leftarrow i + 1
         Swap(A[i], A[i])
   end if
end for
Swap(A[i+1], A[high])
return i + 1
```

Pivot: ☐ i: ☐ j: ☐ Swap: ☐

i=-1

5

4

2

.

### Introduction

Real life Example of Randomness

Deterministic Algorithms

Randomized Algorithm

### Randomized Min

Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

### Classic Quickson and its

### and its Optimization

Quicksort

The Algorithm

Time Complexity

```
\begin{aligned} & \text{pivot} \leftarrow A[high] \\ & \text{i} \leftarrow low - 1 \\ & \text{for } j = low \text{ to } high - 1 \text{ do} \\ & \text{if } A[j] \leq pivot \text{ then} \\ & \text{i} \leftarrow \text{i} + 1 \\ & \text{Swap}(A[i], A[j]) \end{aligned} end if i not getting updated end for & \text{Swap}(A[i+1], A[high]) return i + 1
```



i=-1

5

4

2

1

### Randomized Min

Intro to Min Cut

Randomized Min Cut

### and its

The Algorithm

```
pivot \leftarrow A[high]
i \leftarrow low - 1
for j = low to high - 1 do
   if A[j] < pivot then
         i \leftarrow i + 1
          Swap(A[i], A[i])
    end if i not getting updated
end for
Swap(A[i+1], A[high])
return i + 1
```



i=-1

5

4

2

### Introductio

Real life Example of

Deterministic Algorithms

### Randomized Min

Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

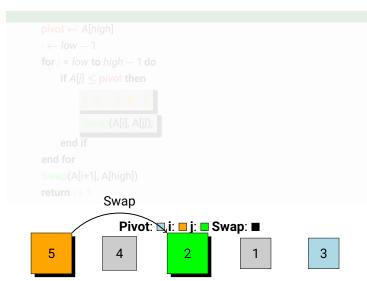
## Classic Quickson

### Optimization

Quicksort

The Algorithm

Time Complexity



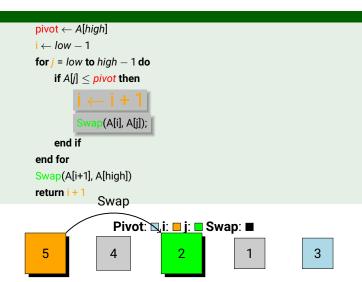
### Randomized Min

Intro to Min Cut

Randomized Min Cut

### and its

The Algorithm



### Introduction

Real life Example (

Deterministic Algorithms

### Randomized Min

Cut Intro to Min Cut

Jaive Algorithm

Randomized Min Cut

### Classic Quickso

### and its Optimization

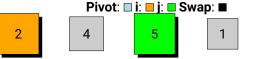
Quicksort

The Algorithm

Time Complexity

Ontimization

```
pivot ← A[high]
i ← low − 1
for j = low to high − 1 do
    if A[j] ≤ pivot then
    i ← i + 1
    Swap(A[i], A[j])
    end if
end for
Swap(A[i+1], A[high])
return i + 1
```



### Introduction

Real life Example of Randomness

Deterministic Algorithms

### Randomized Min

Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

## Classic Quicksor and its

### Optimization

Quicksort

The Algorithm

Time Complexity

```
Swap
Pivot. ■ i: ■ j: ■ Swap: ■
            5
4
```

### Randomized Min

Intro to Min Cut

Randomized Min Cut

## and its

The Algorithm

```
pivot \leftarrow A[high]
i \leftarrow low - 1
for j = low to high - 1 do
    if A[j] < pivot then
          Swap(A[i], A[j]);
    end if
end for
Swap(A[i+1], A[high])
return i + 1
                               Swap
                  Pivot. ■ i: ■ j: ■ Swap: ■
                                  5
                  4
```

### Introduction

Real life Example Randomness

Deterministic Algorithms

### Randomized Min

Cut

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

### Classic Quicks

### and its

Quicksort

The Algorithm

Time Complexity

Ontimization

```
\begin{aligned} & \text{pivot} \leftarrow A[high] \\ & \text{i} \leftarrow low - 1 \\ & \text{for } j = low \text{ to } high - 1 \text{ do} \\ & \text{if } A[j] \leq pivot \text{ then} \\ & \text{i} \leftarrow \text{i} + 1 \\ & \text{Swap}(A[i], A[j]) \\ & \text{end if} \\ & \text{end for} \\ & \text{Swap}(A[i+1], A[high]) \\ & \text{return } \text{i} + 1 \end{aligned}
```



2

1

5

4

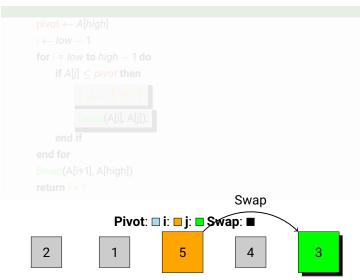
### Randomized Min

Intro to Min Cut

Randomized Min Cut

## and its

The Algorithm



### Randomized Min

Intro to Min Cut

Randomized Min Cut

## and its

The Algorithm

```
pivot \leftarrow A[high]
i \leftarrow low - 1
for j = low to high - 1 do
    if A[j] \leq pivot then
          Swap(A[i], A[j]);
    end if
end for
Swap(A[i+1], A[high])
return i + 1
                                                Swap
                  Pivot: □ i: □ j: □ Swap: ■
                                                   4
```

### Introduction

Real life Example Randomness

Deterministic Algorithms

### Randomized Min

Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

### Classic Quicksor

### and its Optimization

Quicksort

The Algorithm

Time Complexity

Optimization

```
pivot ← A[high]
i ← low − 1
for j = low to high − 1 do
    if A[j] ≤ pivot then
    i ← i + 1
        Swap(A[i], A[j])
    end if
end for
Swap(A[i+1], A[high])
return i + 1
```

Pivot: ■ i: ■ j: ■ Swap: ■

**New Pivot** 

2

1

3

4

### Introduction

Real life Example of Randomness

Deterministic Algorithm Randomized Algorithms

### andomized Mir

Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

## Classic Quicksor and its

### Optimization

The Algorithm

Time Complexit

Time Complexity Optimization

### **QuickSort Partitioning**

 $pivot \leftarrow A[high]$ 

 $\leftarrow low - 1$ 

for j = low to high - 1 do

if A[i] < pivot then

 $i \leftarrow i + 1$ 

Swap(A[i], A[j])

end if

end fo

swap(A[i+1], A[high]

return i + 1

### Continuing QuickSor

Step 1: Pick a pivo

**Step 2:** Partition subarrays

Step 3: Recur for:

- Right Subarray

Repeat until: All elements sorted

Final sorted array: Coming next!

Pivot: ■ i: ■ j: ■ Swap: ■

**New Pivot** 

2

1

3

4

### Introduction

Real life Example of Randomness

Deterministic Algorithm

### andomized Min

Intro to Min Cut

Naive Algorith

Randomized Min Cut

## Classic Quicksort

### Optimization

The Algorithm

Time Complexity

### **QuickSort Partitioning**

```
\begin{aligned} & \underset{\textbf{i}}{\textbf{pivot}} \leftarrow A[high] \\ & \underset{\textbf{i}}{\textbf{i}} \leftarrow low - 1 \\ & \text{for } j = low \text{ to } high - 1 \text{ do} \\ & \underset{\textbf{i}}{\textbf{if}} A[j] \leq pivot \text{ then} \\ & \underset{\textbf{i}}{\textbf{i}} \leftarrow \textbf{i} + 1 \\ & \underset{\textbf{Swap}}{\textbf{Swap}}(A[i], A[j]) \\ & \text{end if} \\ & \text{end for} \\ & \underset{\textbf{Swap}}{\textbf{Swap}}(A[i+1], A[high]) \end{aligned}
```

### Continuing QuickSort

Step 1: Pick a pivot

Step 2: Partition subarrays

Step 3: Recur for:

Left Subarray
 Right Subarray

Repeat until: All elements

Final sorted array: Coming next!

Pivot: ■ i: ■ j: ■ Swap: ■

**New Pivot** 

2

return i + 1

1

3

4

### Introductio

Real life Example of Randomness

Deterministic Algorithm

### andomized Min

Intro to Min Cut

Randomized Min Cut

Nandomized Will Cut

### and its Optimization

### Quicksort

The Algorithm

Time Complexit

### **QuickSort Partitioning**

$$pivot ← A[high]$$

$$i ← low - 1$$

$$for j = low to high - 1 do$$

if 
$$A[j] \leq pivot$$
 then  $i \leftarrow i + 1$ 

Swap(A[i], A[j])

end if

end for

Swap(A[i+1], A[high])

return i + 1

## **Continuing QuickSort**

Step 1: Pick a pivot

Step 2: Partition subarrays

Step 3: Recur for:

- Left Subarray- Right Subarray

Repeat until: All elements

sorted

Final sorted array: Coming next!

Pivot: ■ i: ■ j: ■ Swap: ■

**New Pivot** 

2

1

3

4

### Introduction

Real life Example of

Deterministic Algorithms Randomized Algorithms

### Randomized Min

### Cut

Intro to Min Cut

Naive Algorithm
Randomized Min Cut

## Classic Quicksor and its

### Ouicksort

The Algorithm

Time Complexity

1



**Pivot** 

4

1

2







2

Randomized Algorithms

### Randomized Min

### Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

## and its

The Algorithm

**Pivot** 

### Introduction

Real life Example of Randomness

Deterministic Algorithms
Randomized Algorithms

### Randomized Min

### Cut

Intro to Min Cut

Naive Algorithm Randomized Min Cut

## Classic Quicksor and its

### Quicksort

The Algorithm
Time Complexit

# Pivot

- 2 1
- 3

- 5
- 4

1

2

4

5

2



### Introduction

Real life Example of Randomness

Deterministic Algorithm Randomized Algorithms

### Randomized Min

Intro to Min Cut

Naive Algorithm Randomized Min Cut

### and its Optimization

The Alexan

Time Complexity

Ontimization

$$\begin{array}{c|ccccc}
\hline
1 & 2 & 3 & 4 \\
\hline
& & & & \\
\hline
& & & &$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

**Key Points** 

- Pivot Speciality
  - When!!
- Better idea!!

### Introduction

Real life Example of Randomness

Deterministic Algorithm Randomized Algorithms

### Randomized Min

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

Classic Quickso

### Optimization

The Algor

Time Complexity

Time Complexity

$$\begin{array}{c|cccc}
\hline
1 & 2 & 3 & 4 & 5 \\
\hline
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & \\
\hline
& & & & \\
\hline
&$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

**Key Points** 

- **■** Pivot Speciality
- When!!
- Better idea!!

### Introduction

Real life Example of Randomness

Deterministic Algorithm Randomized Algorithms

## Randomized Min Cut

Intro to Min Cut Naive Algorithm

Randomized Min Cut

### and its Optimization

Quicksort

Time Complexity

Ontimization

$$\begin{array}{c|ccccc}
\hline
1 & 2 & 3 & 4 & 5 \\
\hline
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & & \\$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

### **Key Points**

- Pivot Speciality
- When!!
- Better idea!!

## Randomized Min

Intro to Min Cut Randomized Min Cut

## and its

Time Complexity

$$\begin{array}{c|cccc}
\hline
1 & 2 & 3 & 4 & 5 \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & & \\
& & & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & & & \\
\hline
& & &$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

## Randomized Min

Intro to Min Cut

## and its

Time Complexity

Randomized Min Cut

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

## **Key Points:**

- Pivot Speciality

### Introduction

Real life Example of Randomness

Deterministic Algorithm Randomized Algorithm

## Randomized Min Cut

Intro to Min Cut

Randomized Min Cut

### and its

Quicksort

Time Complexity

Ontimization

$$\begin{array}{c|cccc}
\hline
1 & 2 & 3 & 4 & 5 \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & &$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

## **Key Points:**

- Pivot Speciality
- When!!
- **■** Better idea!!!

### Introduction

Real life Example of Randomness

Deterministic Algorithm
Randomized Algorithm

## Randomized Min Cut

Intro to Min Cut

Randomized Min Cut

### and its

Quicksort

Time Complexity

Ontimization

$$\begin{array}{c|cccc}
\hline
1 & 2 & 3 & 4 & 5 \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & & \\
& & & & \\
\hline
& & & &$$

Total Comparisons = 
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \implies O(n^2)$$

## **Key Points:**

- Pivot Speciality
- When!!
- **■** Better idea!!.

## Could Randomness Help!!

### Introduction

Randomness

Deterministic Algorithm
Randomized Algorithms

### Randomized Min

Intro to Min Cut

Naive Algorithm

Randomized Min Cut

and its

Optimization

The Algorithm

Time Complexity
Optimization

## **Questions to Ponder**

- What if we could make better pivot choices consistently, even for the worst-case scenarios?
- Could randomness help us ensure better performance or average?



## Could Randomness Help!!

Intro to Min Cut

Randomized Min Cut

and its

Optimization

## **Questions to Ponder**

- What if we could make better pivot choices consistently, even for the worst-case scenarios?
- Could randomness help us ensure better performance on average?



## Randomized Quicksort: A Smarter Pivot Choice

### muoductio

## Classic Quickson

Optimization

Quicksor

Time Com

Optimization

Introduction to

### Randomized Quicksort

Randomized Quickso

Algorithm

Deterministic vs

Conclusion

## Key Idea

Random Pivot selection
Avoid Worst-Case partitions

### Objective

- Balanced Partition (on average)
- No Sorted/Reverse worst-case

### How It Works

- Pick Random Pivot
- 2 Swap with last element
- 3 Partition as usual

# Randomized Quicksort: A Smarter Pivot Choice

### introductio

## Classic Quickson

Quicksort

Time Algorith

Time Complex

Introduction to

### Randomized Quicksort

Randomized Quickso

Example

Deterministic vs Randomized Quickso

Conclusion

## Key Idea

Random Pivot selection
Avoid Worst-Case partitions

## **Objective**

- Balanced Partition (on average)
- No Sorted/Reverse worst-case

### **How It Works**

- 1 Pick Random Pivot
- 2 Swap with last element
- 3 Partition as usual

# Randomized Quicksort: A Smarter Pivot Choice

roductio

Classic Quickso

Quicksort

Time Com

Ontimization

Introduction to

Randomized Quicksort Randomized Quicksort

Example Deterministic vs

Randomized Quickso

## Key Idea

Random Pivot selection
Avoid Worst-Case partitions

## **Objective**

- Balanced Partition (on average)
- No Sorted/Reverse worst-case

## **How It Works**

- 1 Pick Random Pivot
- 2 Swap with last element
- 3 Partition as usual

## Randomized QuickSort Algorithm

### Introduction

## Classic Quicksor and its

### Ontimization

Ouicksort

The Algorithm

Time Complexit

Optimization

### Randomized Quicksort Randomized Quicksort

### Randomized Quicksort Algorithm

Deterministic vs Randomized Quicks

Conclusion

# Algorithm QuickSort (Divide and Conquer)

- 1: **Input:** Array A, indices low, high
- 2: Output: Sorted Array
- 3: **if** low < high **then**
- 4: pivot ← Partition(A, low, high)
- 5: QuickSort(A, low, pivot 1)
- 6: QuickSort(A, pivot + 1, high)
- 7: end if

### Algorithm Partition (Pivot Selection)

- 1: **Input:** Array A, indices low, high
- 2: Output: Pivot index
- 3:

```
pivot \leftarrow Random[low, high]
```

- 4:  $i \leftarrow low 1$
- 5: for i = low to high 1 do
- 6: **if**  $A[j] \leq pivot$  **then**
- 7:  $i \leftarrow i+1$
- 8: Swap A[i], A[j]
- 9: **end if**
- 11: Swap A[i + 1] A[high]
- 12: **return** *i* + 1

24/35

## Randomized QuickSort Algorithm

indices

and its

Randomized Quicksort Algorithm

### Algorithm QuickSort (Divide and Conquer)

- 1: **Input:** Array A, low, high
- 2: Output: Sorted Array
- 3: **if** low < high then
- 4: pivot
- **Partition**(A, low, high) QuickSort(A, low, pivot - 1) 5:

QuickSort(A, pivot + 1,

high)

6:

7: end if

## Algorithm Partition (Pivot Selection)

Array A,

indices

24/35

- low, high
- 2: Output: Pivot index 3:
  - $pivot \leftarrow Random[low, high]$

1: **Input:** 

- 4:  $i \leftarrow low 1$
- 5: for j = low to high 1 do if A[i] < pivot then
- $i \leftarrow i + 1$ Swap A[i], A[j]g. end if
- 10: **end for** 11: Swap A[i + 1], A[high]
- 12: **return** *i* + 1

## **Worst Case Review**

### Introduction

## Classic Quicksort and its

**Optimization** 

The Alexandr

Time Complex

Ontimization

Introduction to

Pandomized Quicked

Randomized Quickso

### Example

Deterministic vs Randomized Quickson

### Conclusion

## **Pivot**

1 2









5

2

#### **Worst Case Review**

#### Introduction

# Classic Quicksort and its

Optimization

Quicksoit

Time Complexi

Ontimization

Introduction to

Randomized Quicks

Randomized Quickso Algorithm

#### Example

Deterministic vs Randomized Quickso

#### Conclusion

#### **Pivot**

1 2



4

4

5

1

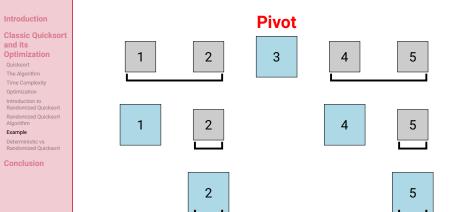
2

5

2

5

#### **Worst Case Review**



Pivot Choice Order: 3 1 2 4 5

#### Introduction

#### Classic Quicksor and its

Ouickeart

Quicksort

Tile Algoridini

Time Complex

Optimization

Introduction to

Randomized Ouickso

Algorithm

Example

Deterministic vs Randomized Quicksort

Conclusion

#### Better Pivot Selection:

- More uniform distribution
- Reduces probability.
- Average Case:
  - Same
  - Avoiding  $O(n^2)$ .

#### Comparison: Deterministic vs Randomized C

#### Introduction

#### Classic Quicksor and its

Ouickeart

Quicksort

The Algorithm

Time Complex

Optimization

Introduction to Randomized Quickso

Randomized Ouicksort

Algorithm

xample

Deterministic vs Randomized Ouicksort

Conclusion

#### **■** Better Pivot Selection:

- More uniform distribution.
- Reduces probability.
- Average Case:
  - Same
  - Avoiding  $O(n^2)$ .

#### Comparison: Deterministic vs Randomized Q

#### Introduction

#### Classic Quicksor and its

Quicksort

The Algorithm

Time Complexit

Ontimization

Introduction to

Randomized Ouickso

Algorithm

xample

Deterministic vs Randomized Quicksort

Conclusion

#### Better Pivot Selection:

- More uniform distribution.
- Reduces probability.
- Average Case:
  - Same.
  - Avoiding  $O(n^2)$ .

#### Comparison: Deterministic vs Randomized Q

#### Introduction

#### Classic Quicksor and its

Quicksort

Time Complexit

Optimization

Introduction to Randomized Quicksort

Randomized Q Algorithm

Algorithm

Deterministic vs

Randomized Quicksort

Conclusion

#### Better Pivot Selection:

- More uniform distribution.
- Reduces probability.
- Average Case:
  - Same.
  - Avoiding  $O(n^2)$ .

#### Comparison: Deterministic vs Randomized Q

Aspect	Deterministic Q	Randomized Q
Pivot Selection	Fixed (e.g., first/last)	Randomly chosen
Worst Case	O(n <sup>2</sup> )	Low probability of $O(n^2)$
Average Case	0(n log n)	$O(n \log n)$

# Food for Thought

Introduction

#### Classic Quicksor and its

Optimization

The Algorithm

Time Complexit

Ontimization

Introduction to Randomized Quickso

Randomized Quickso

xample

Deterministic vs Randomized Quicksort

Conclusion

#### Trade-Off

Probabilistic outcomes.

- Do you think randomization is always beneficial?
- Are there cases where deterministic algorithms outperform randomized ones?



# Food for Thought

Introduction

# Classic Quicksor

Optimizati

Quicksort

Time Orange

Time Complexi

Optimization

Randomized Quicks

Randomized Qu

Algorithm

Example Deterministic vs

Randomized Quicksort

Conclusion

# Trade-Off

Probabilistic outcomes.

- Do you think randomization is always beneficial?
- Are there cases where deterministic algorithms outperform randomized ones?



# Food for Thought

Introduction

# Classic Quicksor and its

Ouicksort

The Algorithm

Time Complexi

Optimization Introduction to

Randomized Quickso

Algorithm Example

Deterministic vs Randomized Quicksort

Conclusion

#### Trade-Off

Probabilistic outcomes.

- Do you think randomization is always beneficial?
- Are there cases where deterministic algorithms outperform randomized ones?



#### The Power of Randomization

Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### Why Randomized Algorithms?

• \$ Use randomness for faster problem-solving.

#### The Power of Randomization

Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### Why Randomized Algorithms?

- Use randomness for faster problem-solving.
- Provide elegant solutions to complexity.

#### The Power of Randomization

Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### Why Randomized Algorithms?

- Use randomness for faster problem-solving.
- Provide elegant solutions to complexity.
- Ensure performance in uncertainty.

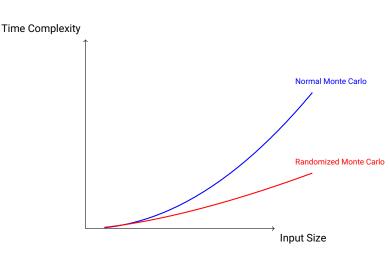
# Comparison of time complexity of Monte Carlo Algorithm

roduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion



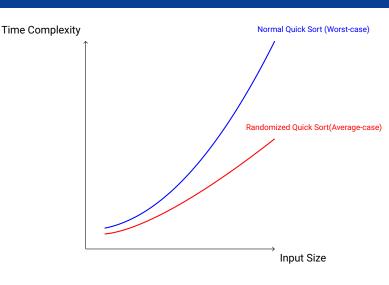
# Comparison of time complexity of Quick Sort

oduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### **Applications**

■ Cryptography: Enhancing security protocols.



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### **Applications**

- Cryptography: Enhancing security protocols.
- L<sup>\*</sup> Optimization: Solving NP-hard problems.



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### **Applications**

- Cryptography: Enhancing security protocols.
- Optimization: Solving NP-hard problems.
- Machine Learning: Improving model accuracy.



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### **Applications**

- Cryptography: Enhancing security protocols.
- Optimization: Solving NP-hard problems.
- Machine Learning: Improving model accuracy.
- Data Analysis: Efficiently processing large datasets.



# Effectiveness of Randomized Algorithms Across Various Domains

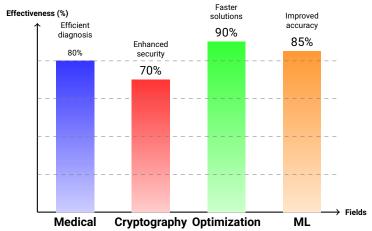
Introduction

randomized Min

Classic Quick Sort and its Optimization

Conclusion





Introduction

randomized Min Cut

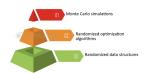
Classic Quick Sort and its Optimization

Conclusion

#### Advantages

 Avoid worst-case outcomes probabilistically.

#### Challenges



Introduction

randomized Min Cut

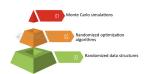
Classic Quick Sort and its Optimization

Conclusion

#### Advantages

- Avoid worst-case outcomes probabilistically.
- <u>Lu</u> Ideal for large-scale problems.

#### Challenges



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### Advantages

- Avoid worst-case outcomes probabilistically.
- Adaptable to various scenarios.

#### Challenges



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

#### Advantages

- Avoid worst-case outcomes probabilistically.
- Ideal for large-scale problems.
- Adaptable to various scenarios.

#### Challenges

Needs high-quality random generators.



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

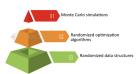
Conclusion

#### Advantages

- Avoid worst-case outcomes probabilistically.
- Ideal for large-scale problems.
- Adaptable to various scenarios.

#### Challenges

- Needs high-quality random generators.
- **☆** Results may vary across runs.



Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

# "In randomness, there is the seed of order."

Someone Inspirational

Introduction

randomized Min Cut

Classic Quick Sort and its Optimization

Conclusion

# Thank You for Your Attention! Questions or

Comments?