Lecture 3: Probabilistic Analysis, Randomized Quicksort

Michael Dinitz

September 7, 2021

601.433/633 Introduction to Algorithms

Introduction: Sorting

- Sorting: given array of comparable elements, put them in sorted order
- Popular topic to cover in Algorithms courses
- This course:
  - I assume you know the basics (mergesort, quicksort, insertion sort, selection sort, bubble sort, etc.) from Data Structures
  - Today: more advanced sorting (randomized quicksort)
  - Next week: Sorting lower bound and ways around it.

Randomized Algorithms and Probabilistic Analysis

First lecture: "Average-case" problematic.

- What is the "average case"?
- Want to design algorithms that work in *all* applications.

Randomized Algorithms and Probabilistic Analysis

First lecture: "Average-case" problematic.

- What is the "average case"?
- Want to design algorithms that work in *all* applications.

Instead of assuming random distribution over inputs (average-case analysis, machine learning), add randomization *inside* algorithm!

- Still assume worst-case inputs, give bound on worst-case *expected* running time.

Randomized Algorithms and Probabilistic Analysis

First lecture: "Average-case" problematic.

- What is the "average case"?
- Want to design algorithms that work in *all* applications.

Instead of assuming random distribution over inputs (average-case analysis, machine learning), add randomization *inside* algorithm!

- Still assume worst-case inputs, give bound on worst-case *expected* running time.

Many Fall semesters: 601.434/634 Randomized and Big Data Algorithms. Great class!

Randomized Algorithms and Probabilistic Analysis

First lecture: "Average-case" problematic.

- What is the "average case"?
- Want to design algorithms that work in *all* applications.

Instead of assuming random distribution over inputs (average-case analysis, machine learning), add randomization *inside* algorithm!

- Still assume worst-case inputs, give bound on worst-case *expected* running time.

Many Fall semesters: 601.434/634 Randomized and Big Data Algorithms. Great class!

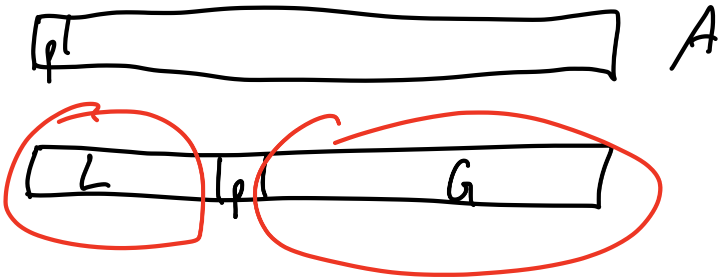
Today: adding randomness into quicksort.

Quicksort Basics (Review)

Input: array **A** of length **n**.

Algorithm:

1. If **n** = **0** or **1**, return **A** (already sorted)
2. Pick some element **p** as the *pivot*
3. Compare every element of **A** to **p**. Let **L** be the elements less than **p**, let **G** be the elements larger than **p**. Create array $[\mathbf{L}, \mathbf{p}, \mathbf{G}]$
4. Recursively sort **L** and **G**.



Quicksort Basics (Review)

Input: array $\mathbf{A}$ of length $\mathbf{n}$.

Algorithm:

1. If $\mathbf{n} = \mathbf{0}$ or $\mathbf{1}$, return $\mathbf{A}$ (already sorted)
2. Pick some element $\mathbf{p}$ as the *pivot*
3. Compare every element of $\mathbf{A}$ to $\mathbf{p}$. Let $\mathbf{L}$ be the elements less than $\mathbf{p}$, let $\mathbf{G}$ be the elements larger than $\mathbf{p}$. Create array $[\mathbf{L}, \mathbf{p}, \mathbf{G}]$
4. Recursively sort $\mathbf{L}$ and $\mathbf{G}$.

Not fully specified: how to choose $\mathbf{p}$?

- Traditionally: some simple deterministic choice (first element, last element, etc.)
- Next lecture: better deterministic choice (not very practical)
- Now: first element

Quicksort Analysis

Upper bound:

If **p** picked as pivot in step 2, then in correct place after step **3**

Quicksort Analysis

Upper bound:

If **p** picked as pivot in step 2, then in correct place after step **3**

$\implies$ step **2** and **3** executed at most **n** times.

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$
$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$
$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)
$\implies$ total time at most $\mathbf{O(n^2)}$

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$
$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)
$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$

$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)

$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.

$\implies$ $\mathbf{p} = \mathbf{A}[0]$ is smallest element

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$

$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)

$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.

$\implies \mathbf{p} = \mathbf{A[0]}$ is smallest element $\implies \mathbf{L} = \emptyset$ and $\mathbf{G} = \mathbf{A[1..n-1]}$

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$

$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)

$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.

$\implies$ $\mathbf{p} = \mathbf{A[0]}$ is smallest element $\implies$ $\mathbf{L} = \emptyset$ and $\mathbf{G} = \mathbf{A[1..n-1]}$

$\implies$ in one call to quicksort, do $\mathbf{\Omega(n)}$ work to compare everything to $\mathbf{p}$, then recurse on array of size $\mathbf{n-1}$

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$
$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)
$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.
$\implies$ $\mathbf{p = A[0]}$ is smallest element $\implies$ $\mathbf{L} = \varnothing$ and $\mathbf{G = A[1..n-1]}$
$\implies$ in one call to quicksort, do $\mathbf{\Omega(n)}$ work to compare everything to $\mathbf{p}$, then recurse on array of size $\mathbf{n-1}$
$\implies$ running time is $\mathbf{T(n) = T(n-1) + cn}$

Quicksort Analysis

Upper bound:

If $\mathbf{p}$ picked as pivot in step 2, then in correct place after step $\mathbf{3}$
$\implies$ step $\mathbf{2}$ and $\mathbf{3}$ executed at most $\mathbf{n}$ times.

Step $\mathbf{3}$ takes time $\mathbf{O(n)}$ (compare every element to pivot)
$\implies$ total time at most $\mathbf{O(n^2)}$

Lower Bound:

Suppose $\mathbf{A}$ already sorted.
$\implies \mathbf{p} = \mathbf{A}[\mathbf{0}]$ is smallest element $\implies \mathbf{L} = \emptyset$ and $\mathbf{G} = \mathbf{A}[\mathbf{1..n-1}]$
$\implies$ in one call to quicksort, do $\mathbf{\Omega(n)}$ work to compare everything to $\mathbf{p}$, then recurse on array of size $\mathbf{n-1}$
$\implies$ running time is $\mathbf{T(n) = T(n-1) + cn} \implies \mathbf{T(n) = \Theta(n^2)}$

Randomized Quicksort

Randomized Quicksort: pick $\mathbf{p}$ *uniformly at random* from $\mathbf{A}$.

Today: prove that *expected* running time at most $\mathbf{O(n \log n)}$ for *every* input $\mathbf{A}$.

Randomized Quicksort

Randomized Quicksort: pick $\mathbf{p}$ *uniformly at random* from $\mathbf{A}$.

Today: prove that *expected* running time at most $\mathbf{O(n \log n)}$ for *every* input $\mathbf{A}$.

- Better than an average-case bound: holds for every single input!
- Maybe in one application inputs tend to be pretty well-sorted: original deterministic quicksort bad, this still good!

Randomized Quicksort

Randomized Quicksort: pick $\mathbf{p}$ *uniformly at random* from $\mathbf{A}$.

Today: prove that *expected* running time at most $\mathbf{O(n \log n)}$ for *every* input $\mathbf{A}$.

- Better than an average-case bound: holds for every single input!
- Maybe in one application inputs tend to be pretty well-sorted: original deterministic quicksort bad, this still good!
- Today only expectation. Can be more clever to get high probability bounds.

Randomized Quicksort

Randomized Quicksort: pick $\mathbf{p}$ *uniformly at random* from $\mathbf{A}$.

Today: prove that *expected* running time at most $\mathbf{O(n \log n)}$ for *every* input $\mathbf{A}$.

- Better than an average-case bound: holds for every single input!
- Maybe in one application inputs tend to be pretty well-sorted: original deterministic quicksort bad, this still good!
- Today only expectation. Can be more clever to get high probability bounds.

Before doing analysis, quick review of basic probability theory.

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

$\boldsymbol{\Omega}$: Sample space. Set of all possible outcomes.

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

$\boldsymbol{\Omega}$: Sample space. Set of all possible outcomes.

- Roll two dice. $\boldsymbol{\Omega} =$

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

$\boldsymbol{\Omega}$: Sample space. Set of all possible outcomes.

- Roll two dice. $\boldsymbol{\Omega} = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{6}\} \times \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{6}\}$.

  $[6] \times [6]$

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

$\boldsymbol{\Omega}$: Sample space. Set of all possible outcomes.

- Roll two dice. $\boldsymbol{\Omega} = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{6}\} \times \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{6}\}$. *Not* $\{\mathbf{2}, \mathbf{3}, \ldots, \mathbf{12}\}$

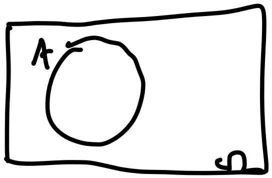
Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

Ω: Sample space. Set of all possible outcomes.

- Roll two dice. $\Omega = \{1, 2, \ldots, 6\} \times \{1, 2, \ldots, 6\}$. *Not* $\{2, 3, \ldots, 12\}$

Event: subset of Ω



Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

Ω: Sample space. Set of all possible outcomes.

- Roll two dice. $\Omega = \{1, 2, \ldots, 6\} \times \{1, 2, \ldots, 6\}$. *Not* $\{2, 3, \ldots, 12\}$

Event: subset of Ω

- "Event that first die is 3": $\{(3, x) : x \in \{1, 2, \ldots, 6\}\}$
- "Event that dice add up to 7 or 11": $\{(x, y) \in \Omega : (x + y = 7) \text{ or } (x + y = 11)\}$

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

Ω: Sample space. Set of all possible outcomes.

- Roll two dice. $\Omega = \{1, 2, \ldots, 6\} \times \{1, 2, \ldots, 6\}$. *Not* $\{2, 3, \ldots, 12\}$

Event: subset of Ω

- "Event that first die is 3": $\{(3, x) : x \in \{1, 2, \ldots, 6\}\}$
- "Event that dice add up to 7 or 11": $\{(x, y) \in \Omega : (x + y = 7) \text{ or } (x + y = 11)\}$

Random Variable: $X : \Omega \to \mathbb{R}$

- X_1: value of first die. $X_1(x, y) = x$
- X_2: value of second die. $X_2(x, y) = y$
- $X = X_1 + X_2$: sum of the dice. $X(x, y) = x + y = X_1(x, y) + X_2(x, y)$

Probability Basics I

Only semi-formal here. Look at CLRS Chapter 5, take Introduction to Probability

Ω: Sample space. Set of all possible outcomes.

- Roll two dice. $\Omega = \{1, 2, \ldots, 6\} \times \{1, 2, \ldots, 6\}$. *Not* $\{2, 3, \ldots, 12\}$

Event: subset of Ω

- "Event that first die is 3": $\{(3, x) : x \in \{1, 2, \ldots, 6\}\}$
- "Event that dice add up to 7 or 11": $\{(x, y) \in \Omega : (x + y = 7) \text{ or } (x + y = 11)\}$

Random Variable: $X : \Omega \to \mathbb{R}$

- X_1: value of first die. $X_1(x, y) = x$
- X_2: value of second die. $X_2(x, y) = y$
- $X = X_1 + X_2$: sum of the dice. $X(x, y) = x + y = X_1(x, y) + X_2(x, y)$

Random variables super important! Running time of randomized quicksort is a random variable.

Probability Basics II

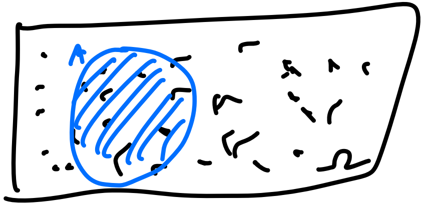
Want to define probabilities. Should use measure theory. Won't.

Probability Basics II

Want to define probabilities. Should use measure theory. Won't.

For each $\mathbf{e} \in \mathbf{\Omega}$ let $\mathbf{Pr[e]}$ be probability of $\mathbf{e}$ (probability distribution)

- $\mathbf{Pr[e] \geq 0}$ for all $\mathbf{e} \in \mathbf{\Omega}$, and $\sum_{\mathbf{e} \in \mathbf{\Omega}} \mathbf{Pr[e] = 1}$
- Probability of an event $\mathbf{A}$ is $\mathbf{Pr[A]} = \sum_{\mathbf{e} \in \mathbf{A}} \mathbf{Pr[e]}$



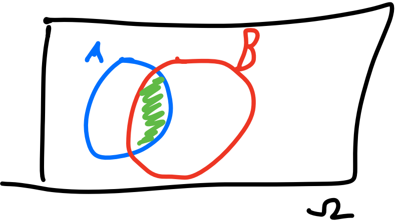
Probability Basics II

Want to define probabilities. Should use measure theory. Won't.

For each $\mathbf{e} \in \mathbf{\Omega}$ let $\mathbf{Pr[e]}$ be probability of $\mathbf{e}$ (probability distribution)

- $\mathbf{Pr[e] \geq 0}$ for all $\mathbf{e} \in \mathbf{\Omega}$, and $\sum_{\mathbf{e} \in \mathbf{\Omega}} \mathbf{Pr[e] = 1}$
- Probability of an event $\mathbf{A}$ is $\mathbf{Pr[A]} = \sum_{\mathbf{e} \in \mathbf{A}} \mathbf{Pr[e]}$

Conditional probability: if $\mathbf{A}$ and $\mathbf{B}$ are events:



$$\mathbf{Pr[B|A]} = \frac{\mathbf{Pr[A \cap B]}}{\mathbf{Pr[A]}} = \frac{\sum_{\mathbf{e} \in \mathbf{A \cap B}} \mathbf{Pr[e]}}{\sum_{\mathbf{e} \in \mathbf{A}} \mathbf{Pr[e]}}$$

Probability Basics III: Expectations

Expectation of a random variable:

$$\mathbf{E}[\mathbf{X}] = \sum_{\mathbf{e} \in \Omega} \mathbf{X}(\mathbf{e}) \mathbf{Pr}[\mathbf{e}]$$

"Average" of the random variable according to probability distribution

Probability Basics III: Expectations

Expectation of a random variable:

$$\mathbf{E}[\mathbf{X}] = \sum_{\mathbf{e} \in \Omega} \mathbf{X}(\mathbf{e}) \mathbf{Pr}[\mathbf{e}]$$

"Average" of the random variable according to probability distribution

Can be useful to rearrange terms to get different equation:

$$\mathbf{E}[\mathbf{X}] = \sum_{\mathbf{e} \in \Omega} \mathbf{X}(\mathbf{e}) \mathbf{Pr}[\mathbf{e}] = \sum_{\mathbf{y} \in \mathbb{R}} \sum_{\mathbf{e} \in \Omega : \mathbf{X}(\mathbf{e}) = \mathbf{y}} \mathbf{y} \cdot \mathbf{Pr}[\mathbf{e}] = \sum_{\mathbf{y} \in \mathbb{R}} \mathbf{y} \cdot \mathbf{Pr}[\mathbf{X} = \mathbf{y}]$$

Probability Basics III: Expectations

Expectation of a random variable:

$$\mathbf{E}[\mathbf{X}] = \sum_{e \in \Omega} \mathbf{X}(e)\mathbf{Pr}[e]$$

"Average" of the random variable according to probability distribution

Can be useful to rearrange terms to get different equation:

$$\mathbf{E}[\mathbf{X}] = \sum_{e \in \Omega} \mathbf{X}(e)\mathbf{Pr}[e] = \sum_{y \in \mathbb{R}} \sum_{e \in \Omega : \mathbf{X}(e) = y} y \cdot \mathbf{Pr}[e] = \sum_{y \in \mathbb{R}} y \cdot \mathbf{Pr}[\mathbf{X} = y]$$

Conditional Expectation: **A** an event, **X** a random variable.

$$\mathbf{E}[\mathbf{X}|\mathbf{A}] = \frac{1}{\mathbf{Pr}[\mathbf{A}]} \sum_{e \in \mathbf{A}} \mathbf{X}(e)\mathbf{Pr}[e]$$

Linearity of Expectations

Amazing feature of expectations: linearity!

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$, *and any constants* α *and* β:

$$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$$

Linearity of Expectations

Amazing feature of expectations: linearity!

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$*, and any constants* α *and* β*:*
$\mathbf{E}[\alpha \mathbf{X} + \beta \mathbf{Y}] = \alpha \mathbf{E}[\mathbf{X}] + \beta \mathbf{E}[\mathbf{Y}]$

Consider rolling two dice. Let $\mathbf{X}$ be sum. What is $\mathbf{E}[\mathbf{X}]$?

- $\mathbf{E}[\mathbf{X}] = \sum_{e \in \Omega} \mathbf{X}(e) \mathbf{Pr}[e]$. 36 term sum!
- $\mathbf{E}[\mathbf{X}] = \sum_{y \in \mathbb{R}} y \cdot \mathbf{Pr}[\mathbf{X} = y]$. What is $\mathbf{Pr}[\mathbf{X} = 2]$, $\mathbf{Pr}[\mathbf{X} = 3]$, ...?

Linearity of Expectations

Amazing feature of expectations: linearity!

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$, *and any constants* α *and* β:
$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$

Consider rolling two dice. Let $\mathbf{X}$ be sum. What is $\mathbf{E}[\mathbf{X}]$?

- $\mathbf{E}[\mathbf{X}] = \sum_{\mathbf{e}\in\Omega}\mathbf{X}(\mathbf{e})\mathbf{Pr}[\mathbf{e}]$. 36 term sum!
- $\mathbf{E}[\mathbf{X}] = \sum_{\mathbf{y}\in\mathbb{R}}\mathbf{y}\cdot\mathbf{Pr}[\mathbf{X}=\mathbf{y}]$. What is $\mathbf{Pr}[\mathbf{X}=\mathbf{2}]$, $\mathbf{Pr}[\mathbf{X}=\mathbf{3}]$, ...?

Instead: $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$. So $\mathbf{E}[\mathbf{X}] = \mathbf{E}[\mathbf{X}_1 + \mathbf{X}_2] = \mathbf{E}[\mathbf{X}_1] + \mathbf{E}[\mathbf{X}_2]$

$X(e) = X_1(e) + X_2(e) \quad \forall e \in \Omega$

Linearity of Expectations

Amazing feature of expectations: linearity!

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$*, and any constants* α *and* β*:*
$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$

Consider rolling two dice. Let $\mathbf{X}$ be sum. What is $\mathbf{E}[\mathbf{X}]$?

- $\mathbf{E}[\mathbf{X}] = \sum_{e \in \Omega} \mathbf{X}(e)\mathbf{Pr}[e]$. 36 term sum!
- $\mathbf{E}[\mathbf{X}] = \sum_{y \in \mathbb{R}} y \cdot \mathbf{Pr}[\mathbf{X} = y]$. What is $\mathbf{Pr}[\mathbf{X} = 2]$, $\mathbf{Pr}[\mathbf{X} = 3]$, ...?

Instead: $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$. So $\mathbf{E}[\mathbf{X}] = \mathbf{E}[\mathbf{X}_1 + \mathbf{X}_2] = \mathbf{E}[\mathbf{X}_1] + \mathbf{E}[\mathbf{X}_2]$

$$\mathbf{E}[\mathbf{X}_1] = \mathbf{E}[\mathbf{X}_2] = \sum_{y=1}^{6} \frac{1}{6} y = \frac{21}{6} = 3.5$$

Linearity of Expectations

Amazing feature of expectations: linearity!

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$, *and any constants* α *and* β*:*
$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$

Consider rolling two dice. Let $\mathbf{X}$ be sum. What is $\mathbf{E}[\mathbf{X}]$?

- $\mathbf{E}[\mathbf{X}] = \sum_{e \in \Omega} \mathbf{X}(e)\mathbf{Pr}[e]$. 36 term sum!
- $\mathbf{E}[\mathbf{X}] = \sum_{y \in \mathbb{R}} y \cdot \mathbf{Pr}[\mathbf{X} = y]$. What is $\mathbf{Pr}[\mathbf{X} = 2]$, $\mathbf{Pr}[\mathbf{X} = 3]$, ...?

Instead: $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$. So $\mathbf{E}[\mathbf{X}] = \mathbf{E}[\mathbf{X}_1 + \mathbf{X}_2] = \mathbf{E}[\mathbf{X}_1] + \mathbf{E}[\mathbf{X}_2]$

$$\mathbf{E}[\mathbf{X}_1] = \mathbf{E}[\mathbf{X}_2] = \sum_{y=1}^{6} \frac{1}{6} y = \frac{21}{6} = 3.5$$

$\implies \mathbf{E}[\mathbf{X}] = 3.5 + 3.5 = 7$

Linearity of Expectations: Proof

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$, *and any constants* α *and* β:

$$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$$

Proof.

$$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \sum_{e \in \Omega} \mathbf{Pr}[e]\,(\alpha\mathbf{X}(e) + \beta\mathbf{Y}(e))$$

Linearity of Expectations: Proof

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$*, and any constants* α *and* β*:*
$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$

Proof.

$$\begin{aligned}\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] &= \sum_{\mathbf{e}\in\Omega} \mathbf{Pr}[\mathbf{e}]\left(\alpha\mathbf{X}(\mathbf{e}) + \beta\mathbf{Y}(\mathbf{e})\right) \\ &= \alpha \sum_{\mathbf{e}\in\Omega} \mathbf{Pr}[\mathbf{e}]\mathbf{X}(\mathbf{e}) + \beta \sum_{\mathbf{e}\in\Omega} \mathbf{Pr}[\mathbf{e}]\mathbf{X}(\mathbf{e})\end{aligned}$$

Linearity of Expectations: Proof

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$, *and any constants* α *and* β*:*
$\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] = \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]$

Proof.

$$\begin{aligned}
\mathbf{E}[\alpha\mathbf{X} + \beta\mathbf{Y}] &= \sum_{e \in \Omega} \mathbf{Pr}[e]\left(\alpha\mathbf{X}(e) + \beta\mathbf{Y}(e)\right) \\
&= \alpha \sum_{e \in \Omega} \mathbf{Pr}[e]\mathbf{X}(e) + \beta \sum_{e \in \Omega} \mathbf{Pr}[e]\mathbf{X}(e) \\
&= \alpha\mathbf{E}[\mathbf{X}] + \beta\mathbf{E}[\mathbf{Y}]
\end{aligned}$$

Linearity of Expectations: Proof

Theorem

For any two random variables $\mathbf{X}$ *and* $\mathbf{Y}$*, and any constants* α *and* β*:*
$\mathbf{E}[\alpha \mathbf{X} + \beta \mathbf{Y}] = \alpha \mathbf{E}[\mathbf{X}] + \beta \mathbf{E}[\mathbf{Y}]$

Proof.

$$\begin{aligned}
\mathbf{E}[\alpha \mathbf{X} + \beta \mathbf{Y}] &= \sum_{e \epsilon \Omega} \Pr[e] (\alpha \mathbf{X}(e) + \beta \mathbf{Y}(e)) \\
&= \alpha \sum_{e \epsilon \Omega} \Pr[e] \mathbf{X}(e) + \beta \sum_{e \epsilon \Omega} \Pr[e] \mathbf{X}(e) \\
&= \alpha \mathbf{E}[\mathbf{X}] + \beta \mathbf{E}[\mathbf{Y}]
\end{aligned}$$

Holds no matter how correlated **X** and **Y** are!

Randomized Quicksort I

Theorem

The expected running time of randomized quicksort is at most $\mathbf{O(n \log n)}$.

Randomized Quicksort I

Theorem

The expected running time of randomized quicksort is at most $\mathbf{O(n \log n)}$.

Assume for simplicity all elements distinct. Running time $= \mathbf{\Theta(}\#$ of comparisons$\mathbf{)}$

Randomized Quicksort I

Theorem

The expected running time of randomized quicksort is at most $\mathbf{O(n \log n)}$.

Assume for simplicity all elements distinct. Running time $= \mathbf{\Theta}(\#$ of comparisons$)$

Definitions:

- $\mathbf{X} = \#$ of comparisons (random variable)
- $\mathbf{e_i} = \mathbf{i}$'th smallest element (for $\mathbf{i} \in \{\mathbf{1}, \dots, \mathbf{n}\}$)
- $\mathbf{X_{ij}}$ random variable for all $\mathbf{i}, \mathbf{j} \in \{\mathbf{1}, \dots, \mathbf{n}\}$ with $\mathbf{i} < \mathbf{j}$:

$$\mathbf{X_{ij}} = \begin{cases} \mathbf{1} & \text{if algorithm compares } \mathbf{e_i} \text{ and } \mathbf{e_j} \text{ at any point in time} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Simple cases:

Randomized Quicksort II

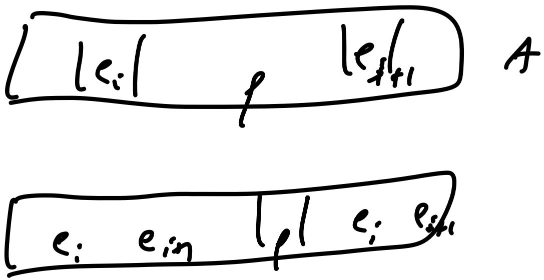
Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Simple cases:

- $\mathbf{j} = \mathbf{i} + \mathbf{1}$:



Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Simple cases:

- $\mathbf{j = i + 1}$: $\mathbf{X_{ij} = 1}$ no matter what, so $\mathbf{E}[\mathbf{X_{ij}}] = \mathbf{1}$

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Simple cases:

- $\mathbf{j = i + 1}$: $\mathbf{X_{ij} = 1}$ no matter what, so $\mathbf{E}[\mathbf{X_{ij}}] = \mathbf{1}$
- $\mathbf{i = 1, j = n}$:

Randomized Quicksort II

Algorithm never compares the same two elements more than once $\implies \mathbf{X} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}$

$$\mathbf{E}[\mathbf{X}] = \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{X_{ij}}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[\mathbf{X_{ij}}]$$

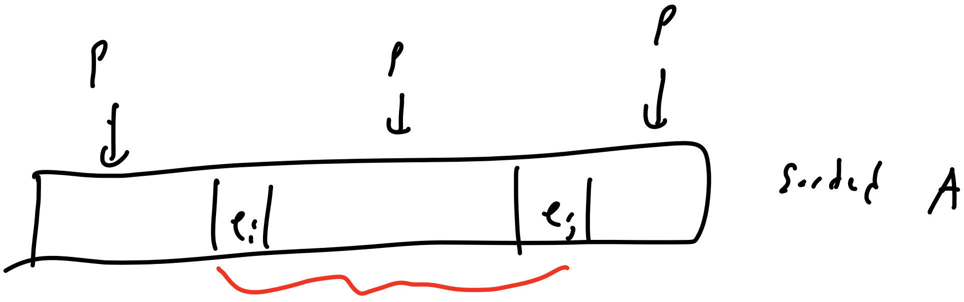
So just need to understand $\mathbf{E}[\mathbf{X_{ij}}]$

Simple cases:

- $\mathbf{j} = \mathbf{i} + \mathbf{1}$: $\mathbf{X_{ij}} = \mathbf{1}$ no matter what, so $\mathbf{E}[\mathbf{X_{ij}}] = \mathbf{1}$
- $\mathbf{i} = \mathbf{1}, \mathbf{j} = \mathbf{n}$: $\mathbf{e_1}$ and $\mathbf{e_n}$ compared if and only if first pivot chosen is $\mathbf{e_1}$ or $\mathbf{e_n}$
  $\implies \mathbf{E}[\mathbf{X_{1n}}] = \frac{\mathbf{2}}{\mathbf{n}}$ $= 1 \cdot \frac{2}{n} + 0 \cdot (1 - \frac{2}{n})$

$\mathbf{E}[\mathbf{X}_{ij}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e}_i$ or $\mathbf{p} = \mathbf{e}_j$:



$X_{ij} = 1$ iff e_i or e_j chosen as pivot before any e_k, $i < k < j$

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$:

$\mathbf{E[X_{ij}]}$: General Case ($\mathbf{i < j}$)

If $\mathbf{p = e_i}$ or $\mathbf{p = e_j}$: $\mathbf{X_{ij} = 1}$

If $\mathbf{e_i < p < e_j}$: $\mathbf{X_{ij} = 0}$

$\mathbf{E[X_{ij}]}$: General Case ($\mathbf{i < j}$)

If $\mathbf{p = e_i}$ or $\mathbf{p = e_j}$: $\mathbf{X_{ij} = 1}$

If $\mathbf{e_i < p < e_j}$: $\mathbf{X_{ij} = 0}$

If $\mathbf{p < e_i}$ or $\mathbf{p > e_j}$:

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij} = 1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij} = 0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

- Condition on $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$:

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

- Condition on $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$: $\mathbf{E}[\mathbf{X_{ij}} \mid \mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}] = \frac{\mathbf{2}}{\mathbf{j-i+1}}$

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

- Condition on $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$: $\mathbf{E}[\mathbf{X_{ij}} \mid \mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}] = \frac{\mathbf{2}}{\mathbf{j-i+1}}$
- Condition on $\mathbf{p} \notin [\mathbf{e_i}, \mathbf{e_j}]$:

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

- Condition on $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$: $\mathbf{E}[\mathbf{X_{ij}} \mid \mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}] = \frac{\mathbf{2}}{\mathbf{j-i+1}}$
- Condition on $\mathbf{p} \notin [\mathbf{e_i}, \mathbf{e_j}]$: still undetermined

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case ($\mathbf{i} < \mathbf{j}$)

If $\mathbf{p} = \mathbf{e_i}$ or $\mathbf{p} = \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{1}$

If $\mathbf{e_i} < \mathbf{p} < \mathbf{e_j}$: $\mathbf{X_{ij}} = \mathbf{0}$

If $\mathbf{p} < \mathbf{e_i}$ or $\mathbf{p} > \mathbf{e_j}$: ? Both $\mathbf{e_i}$, $\mathbf{e_j}$ in same recursive call.

- Condition on $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$: $\mathbf{E}[\mathbf{X_{ij}} \mid \mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}] = \frac{2}{j-i+1}$
- Condition on $\mathbf{p} \notin [\mathbf{e_i}, \mathbf{e_j}]$: still undetermined

So $\mathbf{X_{ij}}$ not determined until $\mathbf{e_i} \leq \mathbf{p} \leq \mathbf{e_j}$, and when it is determined has $\mathbf{E}[\mathbf{X_{ij}}] = \frac{2}{j-i+1}$

$\Longrightarrow \mathbf{E}[\mathbf{X_{ij}}] = \frac{2}{j-i+1}$

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case (formally)

Let $\mathbf{Y_k}$ be event that the $\mathbf{k}$'th pivot is in $[\mathbf{e_i}, \mathbf{e_j}]$ and all previous pivots not in $[\mathbf{e_i}, \mathbf{e_j}]$

$\mathbf{E[X_{ij}]}$: General Case (formally)

Let $\mathbf{Y_k}$ be event that the $\mathbf{k}$'th pivot is in $[\mathbf{e_i}, \mathbf{e_j}]$ and all previous pivots not in $[\mathbf{e_i}, \mathbf{e_j}]$
$\implies$ by definition, the $\mathbf{Y_k}$ events are disjoint and partition sample space

$\mathbf{E}[\mathbf{X_{ij}}]$: General Case (formally)

Let $\mathbf{Y_k}$ be event that the $\mathbf{k}$'th pivot is in $[\mathbf{e_i}, \mathbf{e_j}]$ and all previous pivots not in $[\mathbf{e_i}, \mathbf{e_j}]$
$\implies$ by definition, the $\mathbf{Y_k}$ events are disjoint and partition sample space

Showed that $\mathbf{E}[\mathbf{X_{ij}}|\mathbf{Y_k}] = \frac{2}{\mathbf{j-i+1}}$ for all $\mathbf{k}$.

$E[X_{ij}]$: General Case (formally)

Let Y_k be event that the k'th pivot is in $[e_i, e_j]$ and all previous pivots not in $[e_i, e_j]$
$\implies$ by definition, the Y_k events are disjoint and partition sample space

Showed that $E[X_{ij}|Y_k] = \frac{2}{j-i+1}$ for all k.

$$\begin{aligned} E[X_{ij}] &= \sum_{k=1}^{n} E[X_{ij}|Y_k]\Pr[Y_k] && (Y_k \text{ disjoint and partition } \Omega) \\ &= \frac{2}{j-i+1}\sum_{k=1}^{n}\Pr[Y_k] \\ &= \frac{2}{j-i+1} \end{aligned}$$

Randomized Quicksort: Final Analysis

Expected running time of randomized quicksort:

$$\mathbf{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[X_{ij}] \qquad \text{(linearity of expectations)}$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

$$= 2\sum_{i=1}^{n-1} \left(\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-i+1}\right)$$

$$\leq 2\sum_{i=1}^{n-1} H_n \qquad \left(H_n = \sum_{j=1}^{n} \frac{1}{j}\right)$$

$$\leq 2nH_n$$

$$\leq O(n \log n)$$

$H_n = \Theta(\log n)$