Data Structures and Algorithms
Probability in Computer Science

CS 225
G Carl Evans

April 14, 2025

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
Department of Computer Science

Based on slides by Brad Solomon

Randomization in Algorithms

1. Assume input data is random to estimate average-case performance

2. Use randomness inside algorithm to estimate expected running time

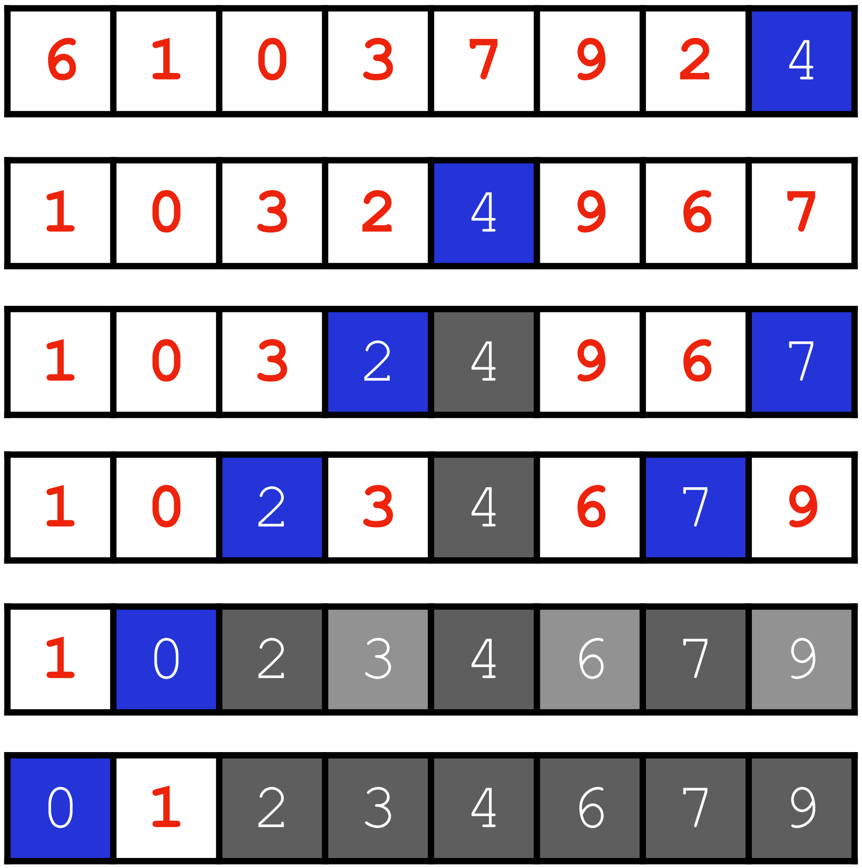
3. Use randomness inside algorithm to approximate solution in fixed tim

Quicksort

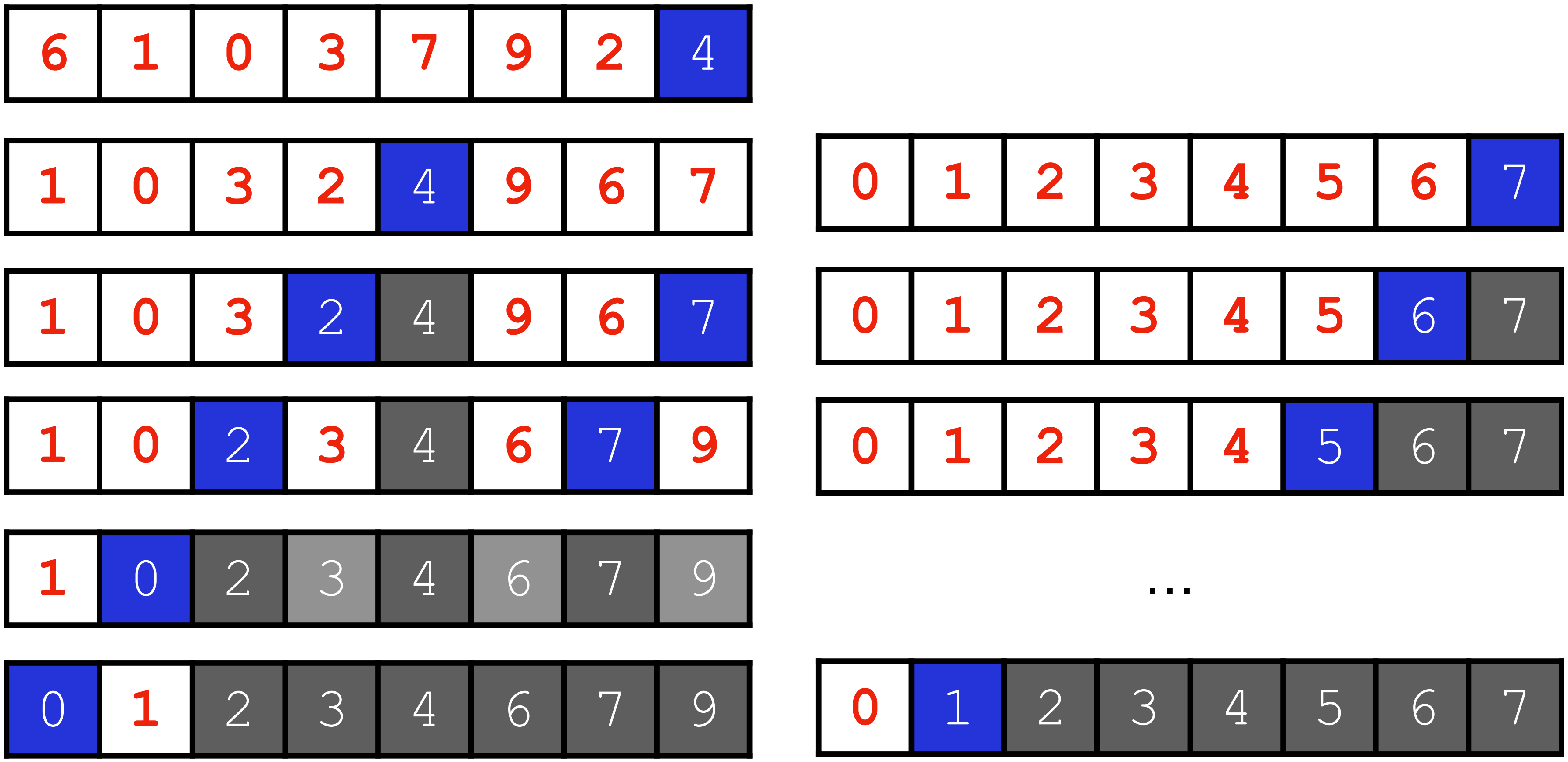
```
Quicksort(A[1…n])
   if n<=1
      return
   p = random(1-n)
   r = Partition(A[1…n],p)
   Quicksort(A[1…r-1])
   Quicksort(A[r+1…n])
```

```
Partition(A[1…n],p)
   swap(A[p],A[n])
   l = 0
   for i 1 to n-1
      if A[i] < A[n]
         l = l + 1
         swap(A[l],A[i])
   l = l + 1
   swap(A[n],A[l])
   return l
```

Expectation Analysis: Randomized Quicksort



Expectation Analysis: Randomized Quicksort



Expectation Analysis: Randomized Quicksort

In **randomized quicksort**, the selection of the pivot is random.

Claim: The expected comparisons is $O(n \log n\, n)$ ***for any input!***

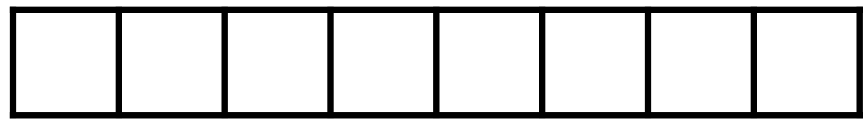
Let X be the total comparisons and X_{ij} be an **indicator variable**:

$$X_{ij} = \begin{cases} 1 \text{if } i\text{th object compared to } j\text{th} \\ 0 \text{ if } i\text{th object not compared to } j\text{th} \end{cases}$$

Then…

Expectation Analysis: Randomized Quicksort

Claim: $E[X_{i,j}] = \frac{2}{j-i+1}$ **Induction:** Assume true for all inputs of $< n$



Expectation Analysis: Randomized Quicksort

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

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

Expectation Analysis: Randomized Quicksort

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

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

$$E[X] = \sum_{i=1}^{n} 2(\frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n-i+1}) \leq 2n \log n$$

Note:

$$H_{n-i+1} - 1 = \left(\frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n - i + 1}\right)$$

$$(H_{n-i+1} - 1) \ \leq \ H_n \leq \log n$$

Expectation Analysis: Randomized Quicksort

Summary: Randomized quick sort is $O(n \log n)$ regardless of input

Randomness:

Assumptions:

Probabilistic Accuracy: Fermat primality test

Pick a random a in the range $[2, p-2]$

If p is prime and a is not divisible by p, then $a^{p-1} \equiv 1 (mod\, p)$

But… ***sometimes*** if n is composite and $a^{n-1} \equiv 1 (mod\, n)$

Probabilistic Accuracy: Fermat primality test

	$a^{p-1} \equiv 1 (mod\, p)$	$a^{p-1} \not\equiv 1 (mod\, p)$
p is prime		
p is not prime		

Probabilistic Accuracy: Fermat primality test

Let's assume $\alpha = .5$

First trial: $a = a_0$ and prime test returns 'prime!'

Second trial: $a = a_1$ and prime test returns 'prime!'

Third trial: $a = a_2$ and prime test returns 'not prime!'

Is our number prime?

What is our **false positive** probability? Our **false negative** probability?