Analysis of Algorithms
CS 477/677

Instructor: Monica Nicolescu

Lecture 9

Midterm Exam

- Tuesday, February 27 in class, 75 minutes
- Exam structure (see practice guide):
  - TRUE/FALSE questions
  - short questions on the topics discussed in class
  - homework-like problems
- All topics discussed up to randomizing quicksort (see lecture 8 slides for Breakpoint 1)
- Mid-term review sessions with Maryam & Jeremy

General Advice for Study

- **Understand** how the algorithms are working
  - Work through the examples we did in class
  - "Narrate" for yourselves the main steps of the algorithms in a few sentences
- Know **when** or **for what problems** the algorithms are applicable
- **Do not memorize** algorithms

Randomized Algorithms

- The behavior is determined in part by values produced by a random-number generator
  - RANDOM(**a, b**) returns an integer **r**, where **a ≤ r ≤ b** and each of the **b-a+1** possible values of **r** is equally likely
- Algorithm generates randomness in input
- No input can consistently elicit worst case behavior
  - Worst case occurs only if we get "unlucky" numbers from the random number generator

Randomized PARTITION

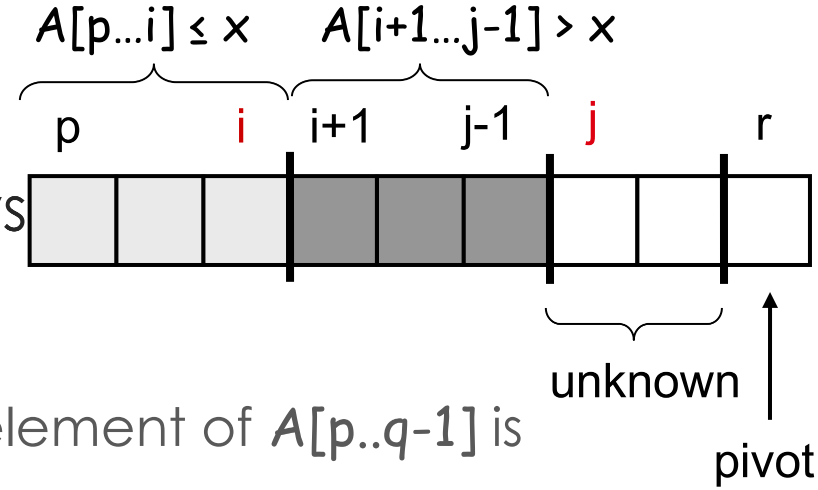
Alg.: RANDOMIZED-PARTITION(A, p, r)

i ← RANDOM(p, r)

exchange A[p] ⟷ A[i]

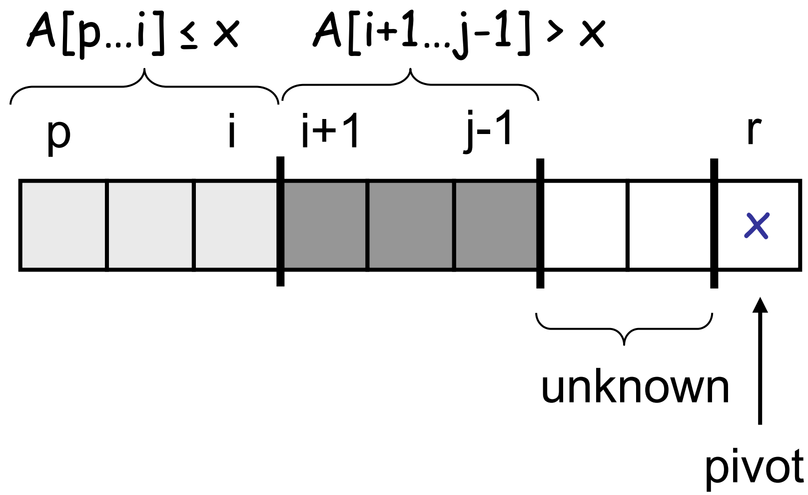
return PARTITION(A, p, r)

Another Way to PARTITION



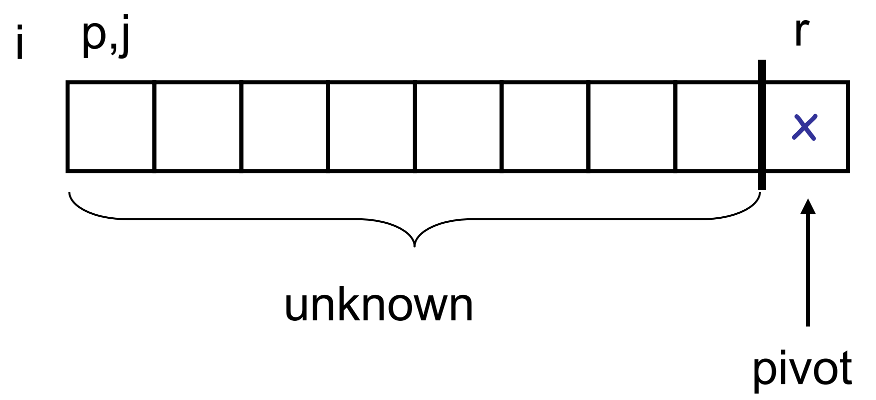
- Given an array ***A***, partition the array into the following subarrays
  - A pivot element ***x*** = ***A[q]***
  - Subarray ***A[p..q-1]*** such that each element of ***A[p..q-1]*** is smaller than or equal to **x** (the pivot)
  - Subarray ***A[q+1..r]***, such that each element of ***A[p..q+1]*** is strictly greater than **x** (the pivot)
- Note: the pivot element is not included in any of the two subarrays

Loop Invariant



1. All entries in $A[p\ ..\ i]$ are smaller than the pivot
2. All entries in $A[i + 1\ ..\ j - 1]$ are strictly larger than the pivot
3. $A[r]$ = pivot
4. $A[j\ ..\ r-1]$ elements not yet examined

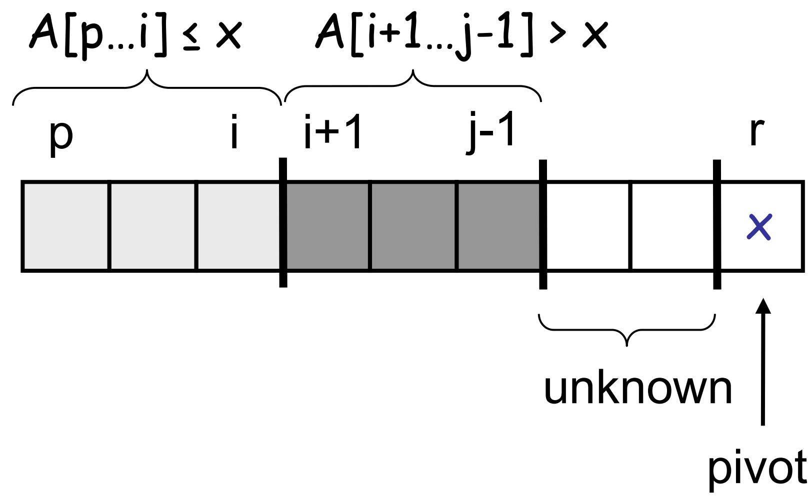
Loop Invariant



Initialization: Before the loop starts:

- $A[r]$ is the pivot
- subarrays $A[p \,..\, i]$ and $A[i + 1 \,..\, j - 1]$ are empty
- All elements in the array are not examined

Loop Invariant



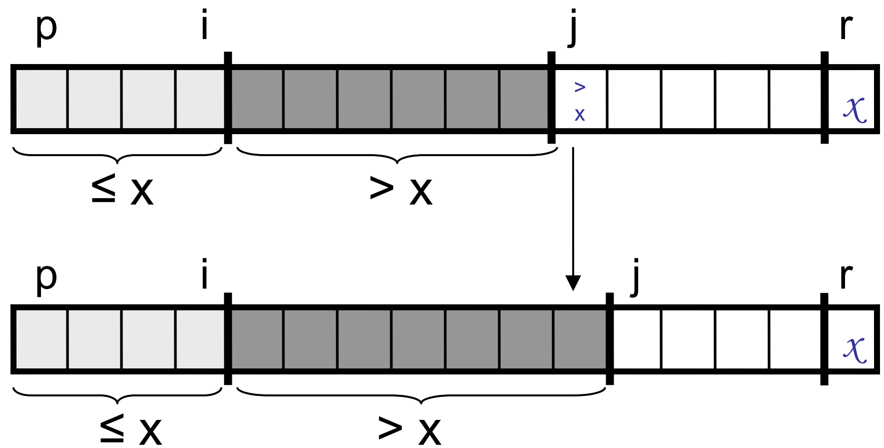
Maintenance: While the loop is running

- if $A[\,j\,] \le$ pivot, then **i** is incremented, $A[\,j\,]$ and $A[i+1]$ are swapped and then j is incremented
- If $A[\,j\,] >$ **pivot**, then increment only j

Maintenance of Loop Invariant

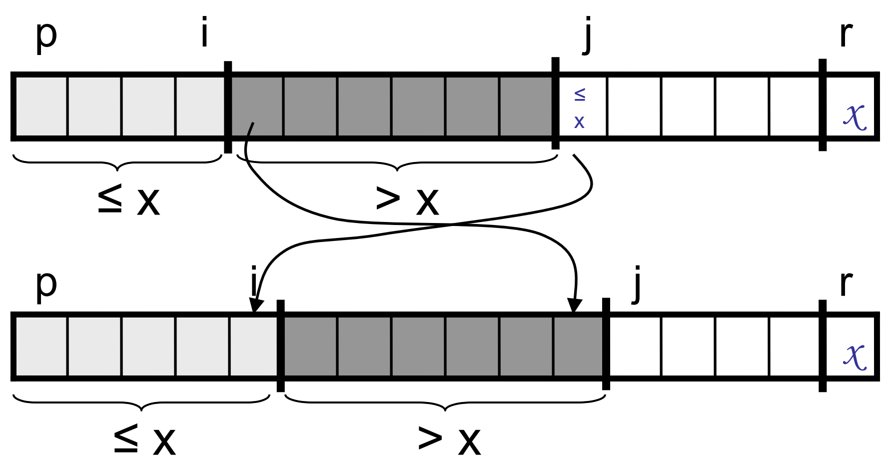
If A[j] > pivot:

- only increment j

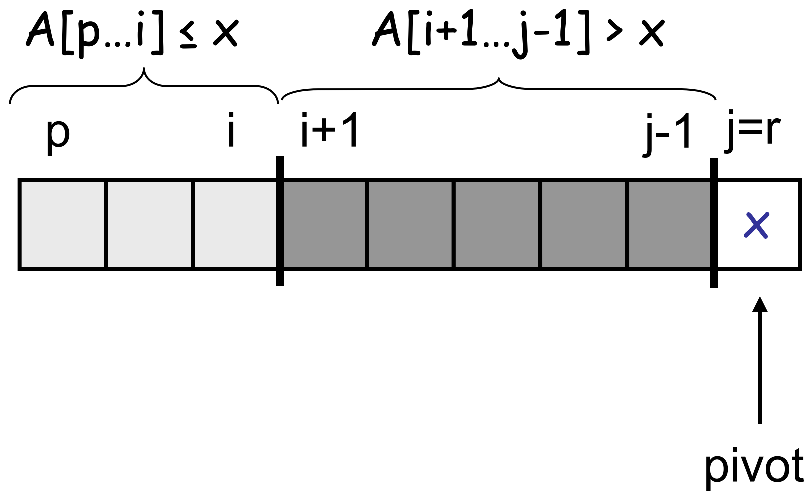


If A[j] ≤ pivot:

- i is incremented, A[j] and A[i] are swapped and then j is incremented



Loop Invariant



Termination: When the loop terminates:

– $j = r \Rightarrow$ all elements in A are partitioned into one of the three cases: $A[p\,..\,i] \le$ pivot, $A[i+1\,..\,r-1] >$ pivot, and $A[r] =$ pivot

Randomized Quicksort

Alg. : RANDOMIZED-QUICKSORT(*A*, p, r)

if p < r

then *q* ← RANDOMIZED-PARTITION2(*A*, p, r)

RANDOMIZED-QUICKSORT(*A*, p, *q* - 1)

The pivot is no longer included in any of the subarrays!!

RANDOMIZED-QUICKSORT(*A*, *q* + 1

Analysis of Randomized Quicksort

Alg. : RANDOMIZED-QUICKSORT(*A*, p, r)

if p < r

The running time of Quicksort is dominated by PARTITION !!

then *q* ← RANDOMIZED-PARTITION2(*A*, p, r)

RANDOMIZED-QUICKSORT(*A*, p, *q* - 1)

RANDOMIZED-QUICKSORT(*A*, *q* + 1, r)

PARTITION is called at most n times

(at each call a pivot is selected and never again included in future calls)

PARTITION

Alg.: PARTITION2*(A, p, r)*

$x \leftarrow A[r]$
$i \leftarrow p - 1$

O(1) - constant

for $j \leftarrow p$ **to** $r - 1$
  do if $A[j] \leq x$
    then $i \leftarrow i + 1$
      exchange $A[i] \leftrightarrow A[j]$

Number of comparisons between the pivot and the other elements

exchange $A[i + 1] \leftrightarrow A[r]$
return $i + 1$

O(1) - constant

Need to compute the total number of comparisons performed in all calls to PARTITION

Random Variables and Expectation

Def.: ***(Discrete) random variable X:*** *a function from a sample space S to the real numbers.*

- It associates a real number with each possible outcome of an experiment

E.g.: X = face of one fair dice

- Possible values: {1, 2, 3, 4, 5, 6}
- Probability to take any of the values: 1/6

Random Variables and Expectation

- Expected value (expectation, mean) of a discrete random variable X is:

$$E[X] = \Sigma_x \, x \, Pr\{X = x\}$$

  – "Average" over all possible values of random variable X

E.g.: X = face of one fair dice

$E[X] = 1\times1/6 + 2\times1/6 + 3\times1/6 + 4\times1/6 + 5\times1/6 + 6\times1/6 = 3.5$

Example

E.g.: flipping two coins:

- Earn \$3 for each head, lose \$2 for each tail
- X: random variable representing your earnings
- Three possible values for variable X:
  - 2 heads $\Rightarrow$ x = \$3 + \$3 = \$6, Pr{2 H's} = ¼
  - 2 tails $\Rightarrow$ x = -\$2 - \$2 = -\$4, Pr{2 T's} = ¼
  - 1 head, 1 tail $\Rightarrow$ x = \$3 - \$2 = \$1, Pr{1 H, 1 T} = ½
- The expected value of X is:

$$E[X] = 6 \times \Pr\{2\ H\text{'s}\} + 1 \times \Pr\{1\ H, 1\ T\} - 4 \times \Pr\{2\ T\text{'s}\}$$

$$= 6 \times \tfrac{1}{4} + 1 \times \tfrac{1}{2} - 4 \times \tfrac{1}{4} = 1$$

More Examples

E.g: X = lottery earnings (15mil. people playing)

- Possible values: 0 and 16.000.000
- Probability to win a 16.000.000 prize: 1/15.000.000
- Probability to win 0: 1 - 1/15.000.000

$$E[X] = 16{,}000{,}000\,\frac{1}{15{,}000{,}000} + 0\,\frac{14{,}999{,}999}{15{,}000{,}000} = \frac{16}{15} = 1.07$$

Indicator Random Variables

- Given a sample space S and an event A, we define the ***indicator random variable*** I{A} associated with A:
  - $I\{A\} = \begin{cases} 1 & \text{if A occurs} \\ 0 & \text{if A does not occur} \end{cases}$
- The expected value of an indicator random variable X_A is: **$E[X_A] = Pr\{A\}$**
- Proof: $E[X_A] = E[I\{A\}] = 1 \times Pr\{A\} + 0 \times Pr\{\bar{A}\} = Pr\{A\}$

Example

- Determine the expected number of heads obtained when flipping a coin
  - Space of possible values: $S = \{H, T\}$
  - Random variable Y: takes on the values H and T, each with probability ½

- Indicator random variable X_H: the coin coming up heads ($Y = H$)
  - Counts the number of heads obtain in the flip
  - $X_H = I\{Y = H\} = \begin{cases} 1 & \text{if } Y = H \\ 0 & \text{if } Y = T \end{cases}$

- The expected number of heads obtained in one flip of the coin is:

$$E[X_H] = E[I\{Y = H\}] = 1 \times \Pr\{Y = H\} + 0 \times \Pr\{Y = T\} =$$

$$= 1 \times \tfrac{1}{2} + 0 \times \tfrac{1}{2} = \tfrac{1}{2}$$

PARTITION

Alg.: PARTITION2(A, p, r)

$x \leftarrow A[r]$
$i \leftarrow p - 1$ } O(1) - constant

for $j \leftarrow p$ **to** $r - 1$

    do if $A[j] \leq x$

        then $i \leftarrow i + 1$

            exchange $A[i] \leftrightarrow A[j]$

Number of comparisons between the pivot and the other elements

exchange $A[i + 1] \leftrightarrow A[r]$
return $i + 1$ } O(1) - constant

Need to compute the total number of comparisons performed in all calls to PARTITION

Number of Comparisons in PARTITION

- Need to compute the **total number of comparisons** performed **in all calls to PARTITION**
- $X_{ij} = I\{z_i \text{ is compared to } z_j\}$
  - For any comparison during the entire execution of the algorithm, not just during one call to PARTITION

When Do We Compare Two Elements?

z_2	z_9	z_8	z_3	z_5	z_4	z_1	z_6	z_{10}	z_7
2	9	8	3	5	4	1	6	10	7

$Z_{1,6} = \{1, 2, 3, 4, 5, 6\}$ $\{7\}$ $Z_{8,10} = \{8, 9, 10\}$

- Rename the elements of A as $z_1, z_2, \ldots, z_n$, with z_i being the i-th smallest element
- Define the set $Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$ the set of elements between z_i and z_j, inclusive

When Do We Compare Elements z_i, z_j?

z_2	z_9	z_8	z_3	z_5	z_4	z_1	z_6	z_{10}	z_7
2	9	8	3	5	4	1	6	10	7

$Z_{1,6} = \{1, 2, 3, 4, 5, 6\}$ $\{7\}$ $Z_{8,10} = \{8, 9, 10\}$

- If pivot x chosen such as: $z_i < x < z_j$
  - z_i and z_j will never be compared
- If z_i or z_j is the pivot
  - z_i and z_j will be compared
  - only if one of them is chosen as pivot before any other element in range z_i to z_j
- Only the pivot is compared with elements in both sets

Number of Comparisons in PARTITION

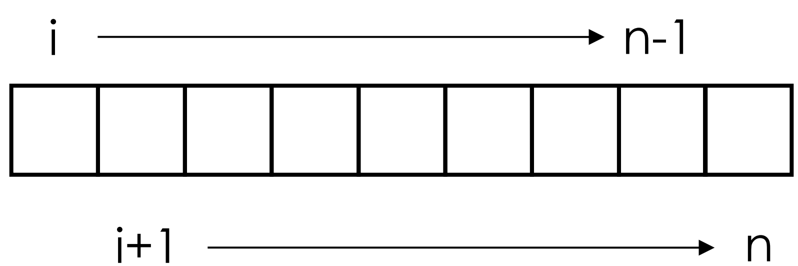
- During the entire run of Quicksort each pair of elements is compared at most once
  - Elements are compared only to the pivot element
  - Since the pivot is never included in future calls to PARTITION, it is never compared to any other element

Number of Comparisons in PARTITION

- Each pair of elements can be compared at most once
  - $X_{ij} = I\{z_i \text{ is compared to } z_j\}$

- Define X as the total number of comparisons performed by the algorithm

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$



Number of Comparisons in PARTITION

- X is an indicator random variable
  - Compute the **expected value**

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

by linearity
of expectation

$$¿\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr\{z_i is compared to z_j\}$$

the expectation of X_{ij} is equal
to the probability of the event
"z_i is compared to z_j"

Number of Comparisons in PARTITION

Pr{ z_i is compared to z_j } =

Pr{ z_i is the first pivot chosen from Z_{ij} }

+

Pr{ z_j is the first pivot chosen from Z_{ij} }

= 1/(j - i + 1) + 1/(j - i + 1) = 2/(j - i + 1)

- There are j – i + 1 elements between z_i and z_j
  - Pivot is chosen randomly and independently
  - The probability that any particular element is the first one chosen is 1/(j - i + 1)

Number of Comparisons in PARTITION

Expected number of comparisons in PARTITION:

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr\{z_i \, is \, compared \, to \, z_j\}$$

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

Change variable: k = j – i ⇒

$$E[X] = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1}$$

We have that: $\sum_{k=1}^{n} \frac{2}{k+1} < \sum_{k=1}^{n} \frac{2}{k}$

$$¿ \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k}$$

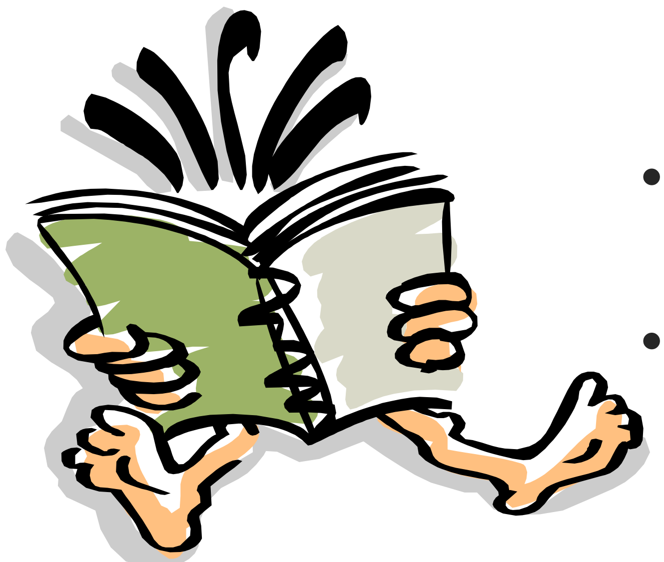
We have that: $\sum_{k=1}^{n} \frac{2}{k} = O(lgn)$

$$¿ \sum_{i=1}^{n-1} O(lgn)$$

=

⇒ Expected running time of Quicksort using RANDOMIZED-PARTITION is O(nlgn)

Readings



- For this lecture
  - Section 7.3-7.4, 9.1, 9.2
- Coming next
  - Section 9.3
  - Chapter 8