CS473 - Algorithms I

Lecture 6-b
Randomized Quicksort

View in slide-show mode

Cevdet Aykanat and Mustafa Ozdal
Computer Engineering Department, Bilkent University

Randomized Quicksort

- In the avg-case analysis, we assumed that all permutations of the input array are equally likely.
  - But, this assumption does not always hold
  - e.g. What if all the input arrays are reverse sorted?

    ➔ Always worst-case behavior

- Ideally, the avg-case runtime should be independent of the input permutation.
- Randomness should be within the algorithm, not based on the distribution of the inputs.

  i.e. The avg case should hold for all possible inputs

Randomized Algorithms

- Alternative to assuming a uniform distribution:

  ➔ Impose a uniform distribution

  e.g. Choose a random pivot rather than the first element

- Typically useful when:
  - there are many ways that an algorithm can proceed
  - but, it's difficult to determine a way that is always guaranteed to be good.
  - If there are many good alternatives; simply choose one randomly.

Randomized Algorithms

- Ideally:
  - Runtime should be independent of the specific inputs
  - No specific input should cause worst-case behavior
  - Worst-case should be determined only by output of a random number generator.

Randomized Quicksort

Using Hoare's partitioning algorithm:

```
R-QUICKSORT(A, $p$, $r$)
    if $p < r$ then
        $q \leftarrow$ R-PARTITION(A, $p$, $r$)
        R-QUICKSORT(A, $p$, $q$)
        R-QUICKSORT(A, $q+1$, $r$)
```

```
R-PARTITION(A, $p$, $r$)
    $s \leftarrow$ RANDOM($p$, $r$)
    exchange $A[p] \leftrightarrow A[s]$
    return H-PARTITION(A, $p$, $r$)
```

Alternatively, permuting the whole array would also work

➔ but, would be more difficult to analyze

Randomized Quicksort

Using Lomuto's partitioning algorithm:

```
R-QUICKSORT(A, p, r)
    if p < r then
        q ← R-PARTITION(A, p, r)
        R-QUICKSORT(A, p, q-1)
        R-QUICKSORT(A, q+1, r)
```

```
R-PARTITION(A, p, r)
    s ← RANDOM(p, r)
    exchange A[r] ↔ A[s]
    return L-PARTITION(A, p, r)
```

Alternatively, permuting the whole array would also work

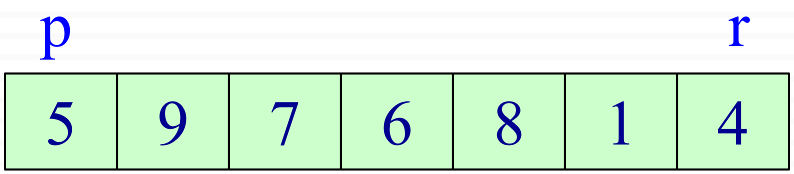
➔ but, would be more difficult to analyze

Notations for Formal Analysis

- Assume all elements in $A[p..r]$ are distinct
- Let $n = r – p + 1$

- Let $rank(x) = \left|\{A[i]: p \leq i \leq r \text{ and } A[i] \leq x\}\right|$

  i.e. rank(x) is the number of array elements with value less than or equal to x



p						r
5	9	7	6	8	1	4

$rank(5) = 3$

i.e. it is the **3**$^{\text{rd}}$ smallest element in the array

Formal Analysis for Average Case

- The following analysis will be for Quicksort using Hoare's partitioning algorithm.
- *Reminder*: The pivot is selected randomly and exchanged with A[p] before calling H-PARTITION

- Let x be the random pivot chosen.
- What is the probability that rank(x) = i for i = 1, 2, …n ?

$$P(\text{rank}(x) = i) = 1/n$$

Various Outcomes of H-PARTITION

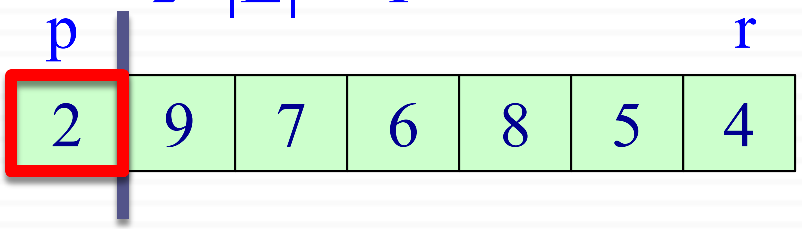
Assume that rank(x) = 1

i.e. the random pivot chosen is the smallest element

What will be the size of the left partition (|L|)?

Reminder: Only the elements less than or equal to x will be in the left partition.

→ |L| = 1



2	9	7	6	8	5	4

pivot = x = 2

Various Outcomes of H-PARTITION

Assume that rank(x) > 1

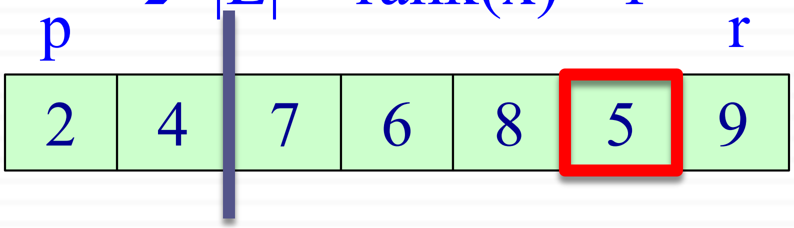
i.e. the random pivot chosen is not the smallest element

What will be the size of the left partition (|L|)?

Reminder: Only the elements less than or equal to x will be in the left partition.

Reminder: The pivot will stay in the right region after H-PARTITION if rank(x) > 1

➔ |L| = rank(x) - 1



pivot = x = 5

Various Outcomes of H-PARTITION - Summary

P(rank(x) = i) = 1/n for $1 \le i \le n$

x: pivot

|L|: size of left region

if rank(x) = 1 **then** |L| = 1

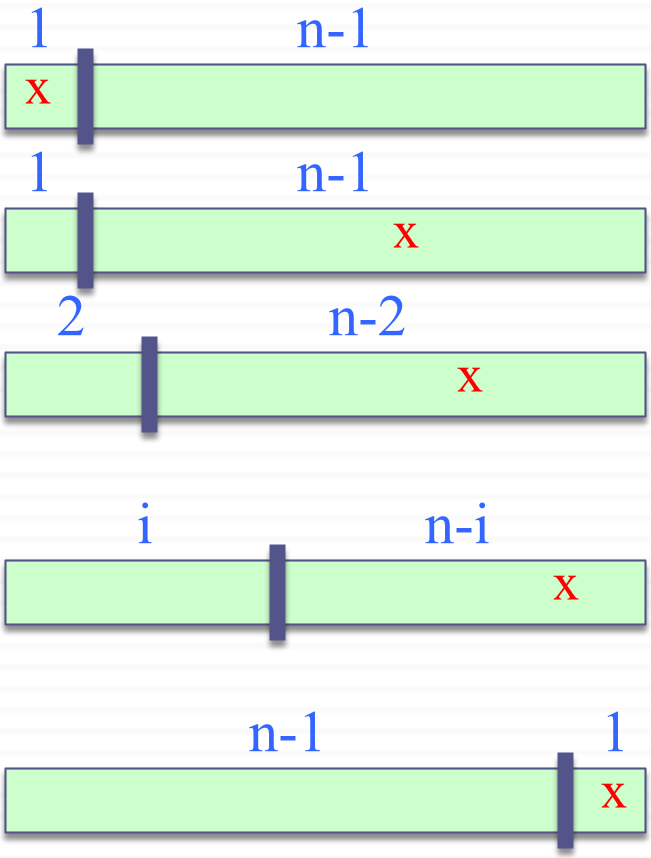
if rank(x) > 1 **then** |L| = rank(x) - 1

P(|L| = 1) = **P**(rank(x) = 1) + **P**(rank(x) = 2) ⇒ **P**(|L| = 1) = 2/n

P(|L| = i) = **P**(rank(x) = i+1) for $1 < i < n$ ⇒ **P**(|L| = i) = 1/n for $1 < i < n$

Various Outcomes of H-PARTITION - Summary

rank(x)	probability	T(n)
1	1/n	$T(1) + T(n-1) + \Theta(n)$
2	1/n	$T(1) + T(n-1) + \Theta(n)$
3	1/n	$T(2) + T(n-2) + \Theta(n)$
⋮	⋮	⋮
i+1	1/n	$T(i) + T(n-i) + \Theta(n)$
⋮	⋮	⋮
n	1/n	$T(n-1) + T(1) + \Theta(n)$



Average - Case Analysis: Recurrence

$$\begin{array}{rllc}
 & & & \underline{rank(x)} \\
T(n) & = & \frac{1}{n}(T(1)+T(n-1)) & 1 \\
 & + & \frac{1}{n}(T(1)+T(n-1)) & 2 \\
 & + & \frac{1}{n}(T(2)+T(n-2)) & 3 \\
 & \vdots & \qquad\vdots & \vdots \\
 & + & \frac{1}{n}(T(\mathrm{i})+T(n-\mathrm{i})) & i+1 \\
 & \vdots & \qquad\vdots & \vdots \\
 & + & \frac{1}{n}(T(\mathrm{n\text{-}1})+T(1)) & n \\
 & + & \Theta(n) &
\end{array}$$

x = pivot

Recurrence

$$T(n) = \frac{1}{n} \sum_{q=1}^{n-1} (T(q)+T(n\text{-}q)) + \frac{1}{n}(T(1)+T(n-1)) + \Theta(n)$$

Note: $\frac{1}{n}(T(1)+T(n\text{-}1)) = \frac{1}{n}(\Theta(1)+ O(n^2)) = O(n)$

$$\Rightarrow \quad T(n) = \frac{1}{n} \sum_{q=1}^{n-1} (T(q)+T(n\text{-}q)) + \Theta(n)$$

- for $k = 1,2,\ldots,n-1$ each term $T(k)$ appears twice

  once for $q = k$ and once for $q = n-k$

- $T(n) = \frac{2}{n} \sum_{k=1}^{n-1} T(k) + \Theta(n)$

Solving Recurrence: Substitution

Guess: $T(n)=O(n\lg n)$

I.H. : $T(k) \le ak\lg k$ *for* $k<n$, for some constant $a > 0$

$$T(n) = \frac{2}{n}\sum_{k=1}^{n-1} T(k) + \Theta(n)$$

$$\le \frac{2}{n}\sum_{k=1}^{n-1} (ak\lg k) + \Theta(n)$$

$$= \frac{2a}{n}\sum_{k=1}^{n-1} (k\lg k) + \Theta(n)$$

Need a tight bound for $\sum k\lg k$

Tight bound for $\sum k\lg k$

- Bounding the terms

$$\sum_{k=1}^{n-1} k\lg k \leq \sum_{k=1}^{n-1} n\lg n = n(n-1)\lg n \leq n^2 \lg n$$

This bound is not strong enough because

- $T(n) \leq \frac{2a}{n} n^2 \lg n + \Theta(n)$

  $= \quad 2an\lg n + \Theta(n)$ → couldn't prove $T(n) \leq an\lg n$

Tight bound for $\sum k\lg k$

- Splitting summations: ignore ceilings for simplicity

$$\sum_{k=1}^{n-1} k\lg k \leq \sum_{k=1}^{n/2-1} k\lg k + \sum_{k=n/2}^{n-1} k\lg k$$

First summation: $\lg k < \lg(n/2) = \lg n - 1$

Second summation: $\lg k < \lg n$

Splitting: $\sum_{k=1}^{n-1} k \lg k \leq \sum_{k=1}^{n/2-1} k \lg k + \sum_{k=n/2}^{n-1} k \lg k$

$$\sum_{k=1}^{n-1} k \lg k \leq (\lg n - 1) \sum_{k=1}^{n/2-1} k + \lg n \sum_{k=n/2}^{n-1} k$$

$$= \lg n \sum_{k=1}^{n-1} k - \sum_{k=1}^{n/2-1} k = \frac{1}{2} n(n-1) \lg n - \frac{1}{2} \frac{n}{2} (\frac{n}{2} - 1)$$

$$= \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2 - \frac{1}{2} n(\lg n - 1/2)$$

$$\boxed{\sum_{k=1}^{n-1} k \lg k \leq \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2} \quad for \lg n \geq 1/2 \Rightarrow n \geq \sqrt{2}$$

Substituting: $\sum_{k=1}^{n-1} k \lg k \le \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2$

$$T(n) \le \frac{2a}{n} \sum_{k=1}^{n-1} k \lg k + \Theta(n)$$

$$\le \frac{2a}{n} \left(\frac{1}{2} n^2 \lg n - \frac{1}{8} n^2\right) + \Theta(n)$$

$$= a n \lg n - \left(\frac{a}{4} n - \Theta(n)\right)$$

We can choose *a* large enough so that $\frac{a}{4} n \ge \Theta(n)$

$$\Rightarrow T(n) \le a n \lg n \Rightarrow \boxed{T(n) = O(n \lg n)}$$ Q.E.D.