Lecture 6-b
Randomized Quicksort

View in slide-show mode
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:

\[\text{R-QUICKSORT}(A, p, r)\]

\[\text{if } p < r \text{ then}\]

\[q \leftarrow \text{R-PARTITION}(A, p, r)\]

\[\text{R-QUICKSORT}(A, p, q)\]

\[\text{R-QUICKSORT}(A, q+1, r)\]

\[\text{R-PARTITION}(A, p, r)\]

\[s \leftarrow \text{RANDOM}(p, r)\]

\[\text{exchange } A[p] \leftrightarrow A[s]\]

\[\text{return } \text{H-PARTITION}(A, p, r)\]

Alternatively, permuting the whole array would also work
\[\rightarrow\text{ but, would be more difficult to analyze}\]
Randomized Quicksort

Using Lomuto’s partitioning algorithm:

$$\text{R-QUICKSORT}(A, p, r)$$

if \( p < r \) then

\[
q \leftarrow \text{R-PARTITION}(A, p, r) \\
\text{R-QUICKSORT}(A, p, q-1) \\
\text{R-QUICKSORT}(A, q+1, r)
\]

$$\text{R-PARTITION}(A, p, r)$$

\[
s \leftarrow \text{RANDOM}(p, r) \\
\text{exchange } A[r] \leftrightarrow A[s] \\
\text{return } \text{L-PARTITION}(A, p, r)
\]

Alternatively, permuting the whole array would also work

\( \Rightarrow \) 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 $\text{rank}(x) = \left| \{A[i]: p \leq i \leq r \text{ and } A[i] \leq x\} \right|$ 
  i.e. $\text{rank}(x)$ is the number of array elements with value less than or equal to $x$

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>9</th>
<th>7</th>
<th>6</th>
<th>8</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank(5) = 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  i.e. it is the 3rd 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 \(\text{rank}(x) = i\) for \(i = 1, 2, \ldots, n\) ?

  \[
P(\text{rank}(x) = i) = \frac{1}{n}
\]
Various Outcomes of H-PARTITION

Assume that $\text{rank}(x) = 1$

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.

$\Rightarrow |L| = 1$

pivot = $x = 2$

| 2 | 9 | 7 | 6 | 8 | 5 | 4 |
Various Outcomes of H-PARTITION

Assume that \( \text{rank}(x) > 1 \)

\( i.e. \) 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 \( \text{rank}(x) > 1 \)

\[ |L| = \text{rank}(x) - 1 \]

pivot = \( x = 5 \)
Various Outcomes of H-PARTITION - Summary

\[ P(\text{rank}(x) = i) = \frac{1}{n} \text{ for } 1 \leq i \leq n \]

\[ \text{if } \text{rank}(x) = 1 \text{ then } |L| = 1 \]

\[ \text{if } \text{rank}(x) > 1 \text{ then } |L| = \text{rank}(x) - 1 \]

\[ P(|L| = 1) = P(\text{rank}(x) = 1) + P(\text{rank}(x) = 2) \]

\[ P(|L| = i) = P(\text{rank}(x) = i+1) \text{ for } 1 < i < n \]

\[ P(|L| = 1) = \frac{2}{n} \]

\[ P(|L| = i) = \frac{1}{n} \text{ for } 1 < i < n \]

x: pivot

|L|: size of left region
### Various Outcomes of H-PARTITION - Summary

<table>
<thead>
<tr>
<th>rank(x)</th>
<th>probability</th>
<th>T(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/n</td>
<td>T(1) + T(n-1) + Θ(n)</td>
</tr>
<tr>
<td>2</td>
<td>1/n</td>
<td>T(1) + T(n-1) + Θ(n)</td>
</tr>
<tr>
<td>3</td>
<td>1/n</td>
<td>T(2) + T(n-2) + Θ(n)</td>
</tr>
<tr>
<td>i+1</td>
<td>1/n</td>
<td>T(i) + T(n-i) + Θ(n)</td>
</tr>
<tr>
<td>n</td>
<td>1/n</td>
<td>T(n-1) + T(1) + Θ(n)</td>
</tr>
</tbody>
</table>

Diagram: Various outcomes of H-PARTITION with intervals and probabilities.
Average - Case Analysis: Recurrence

\[ T(n) = \frac{1}{n} (T(1) + T(n-1)) + \frac{1}{n} (T(1) + T(n-1)) + \frac{1}{n} (T(2) + T(n-2)) + \cdots + \frac{1}{n} (T(i) + T(n-i)) + \cdots + \frac{1}{n} (T(n-1) + T(1)) + \Theta(n) \]

\[ rank(x) \]

\[ x = \text{pivot} \]

\[ n \]

\[ i+1 \]

\[ n \]
Recurrence

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

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

\[ \Rightarrow T(n) = \frac{1}{n} \sum_{q=1}^{n-1} (T(q)+T(n-q)) + \Theta(n) \]

- 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: \( T(n) = O(n \log n) \)

I.H. : \( T(k) \leq ak \log k \) for \( k < n \), for some constant \( a > 0 \)

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

\[
\leq \frac{2}{n} \sum_{k=1}^{n-1} (ak \log k) + \Theta(n)
\]

\[
= \frac{2a}{n} \sum_{k=1}^{n-1} (k \log k) + \Theta(n)
\]

Need a tight bound for \( \sum k \log k \)
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

- $T(n) \leq \frac{2a}{n^2} n^2 \lg n + \Theta(n)$
  
  $$= 2an \lg n + \Theta(n) \quad \Rightarrow \text{couldn’t prove } T(n) \leq an \lg n$$
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: \[ \sum_{k=1}^{n-1} k \log k \leq \sum_{k=1}^{n/2-1} k \log k + \sum_{k=n/2}^{n-1} k \log k \]

\[ \sum_{k=1}^{n-1} k \log k \leq (\log (n-1)) \sum_{k=1}^{n/2-1} k + \log n \sum_{k=n/2}^{n-1} k \]

\[ = \log n \sum_{k=1}^{n-1} k - \sum_{k=1}^{n/2-1} k = \frac{1}{2} n(n-1) \log n - \frac{1}{2} n \left( \frac{n}{2} - 1 \right) \]

\[ = \frac{1}{2} n^2 \log n - \frac{1}{8} n^2 - \frac{1}{2} n (\log n - 1 / 2) \]

\[ \sum_{k=1}^{n-1} k \log k \leq \frac{1}{2} n^2 \log n - \frac{1}{8} n^2 \quad \text{for } \log n \geq 1 / 2 \Rightarrow n \geq \sqrt{2} \]
Substituting: \[
\sum_{k=1}^{n-1} k \lg k \leq \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2
\]

\[
T(n) = \frac{2a}{n} \frac{n^1}{k=1} k \lg k + \quad (n)
\]

\[
= an \lg n \quad \frac{a}{4} n \quad (n) ÷
\]

We can choose \( a \) large enough so that \( \frac{a}{4} n \quad (n) \)

\[
T(n) = an \lg n \quad T(n) = O(n \lg n)
\]

Q.E.D.