Quicksort
Where Average and Worst Case Differ

S.V.N. (vishy) Vishwanathan

University of California, Santa Cruz
vishy@ucsc.edu

February 1, 2016
Outline

1. Basic Idea
2. Worst Case Analysis
3. Average Case Analysis
Divide and Conquer

- **Divide:** Choose a random pivot $a \in A$ and split the array into $A_{<}$ and $A_{>}$.
- **Conquer:** Sort $A_{<}$ and $A_{>}$ by calling quicksort
- **Combine:** Nothing to do! (note that $a$ is already in its correct position)
Pseudocode

QUICKSORT\((A, p, r)\)

1 \textbf{if} \ p < r \\
2 \hspace{1em} q = \text{PARTITION}(A, p, r) \\
3 \hspace{1em} \text{QUICKSORT}(A, p, q - 1) \\
4 \hspace{1em} \text{QUICKSORT}(A, q + 1, r)
Basic Idea

Partition

\[
\begin{array}{ccc}
& i & p \ j \\
(a) & 2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p \ i & j \\
(b) & 2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p \ i & j \\
(c) & 2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p \ i & j \\
(d) & 2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p & i & j \\
(e) & 2 & 1 & 7 & 8 & 3 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p & i & j \\
(f) & 2 & 1 & 3 & 8 & 7 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p & i & j \\
(g) & 2 & 1 & 3 & 8 & 7 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p & i \\
(h) & 2 & 1 & 3 & 8 & 7 & 5 & 6 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& p & i \\
(i) & 2 & 1 & 3 & 4 & 7 & 5 & 6 & 8 \\
\end{array}
\]
Basic Idea

Partition

```
PARTITION(A, p, r)
1   x = A[r]
2   i = p - 1
3   for j = p to r - 1
4       if A[j] ≤ x
5           i = i + 1
6       exchange A[i] with A[j]
7   exchange A[i + 1] with A[r]
8   return i + 1
```
At the beginning of each iteration of the loop of lines 3–6, for any array index $k$

1. If $p \leq k \leq i$, then $A[k] \leq x$.
2. If $i + 1 \leq k \leq j - 1$, then $A[k] > x$.
3. If $k = r$, then $A[k] = x$.

Indices between $j$ and $r - 1$ have no particular relationship to $x$. 
In Pictures

\[ p \leq x < i \quad > x \quad j \quad \text{unrestricted} \quad r \]

\[ p \quad i \quad j \quad r \]

\[ x \]
Initialization

- $i = p - 1$ and $j = p$
- No values lie between $i + 1$ and $j - 1$
- First two conditions are trivially satisfied
- $x = A[r]$ in line 1 satisfies the third condition
Maintenance

(a)

(b)
Maintenance

- **Case 1:** $A[j] > x$
  - Increment $j$
  - Condition 2 now holds for $A[j-1]$
  - Other entries remain unchanged

- **Case 2:** $A[j] \leq x$
  - Increment $i$ and swap $A[i]$ with $A[j]$
  - This satisfies condition 1
  - Increment $j$
  - Now $A[j-1] > x$, this satisfies condition 3
Termination

- $j = r$ so every element in the array is either in $A_{\leq}$ or $A_{>}$
Outline

1. Basic Idea
2. Worst Case Analysis
3. Average Case Analysis
Worst Case Time Complexity

Partitioning procedure produces one sub-problem with \( n - 1 \) elements and one with 0 elements.

Recurrence relationship

\[
T(n) = T(n-1) + T(0) + \Theta(n)
\]

\[
= T(n-1) + \Theta(n)
\]

Therefore \( T(n) = O(n^2) \)

Worst case occurs when the array is already sorted!
Best Case Time Complexity

- Partitioning procedure produces one sub-problem with \( \lfloor n/2 \rfloor \) elements and one with \( \lceil n/2 \rceil - 1 \) elements.
- Recurrence relationship
  \[
  T(n) = 2T(n/2) + \Theta(n)
  \]

Therefore \( T(n) = \Omega(n \log(n)) \)
Outline

1. Basic Idea
2. Worst Case Analysis
3. Average Case Analysis
Average Case

- Suppose each permutation of an array is equally likely as an input to **QUICKSORT**
- What is the expected time complexity?

Turns out the average case time complexity of **QUICKSORT** is $O(n \log n)$
What about Imbalanced Splits?

- Even if the partitioning procedure produces one sub-problem with $9n/10$ elements and one with $n/10$ elements
- Recurrence relationship

\[ T(n) = T(9n/10) + T(n/10) + \Theta(n) \]

Still evaluates to $T(n) = O(n \log(n))$
Average Case Analysis

Proof by Picture

\[ O(n \log n) \]
Randomizing Pivot Selection

RANDOMIZED-PARTITION\((A, p, r)\)
1 \( i = \text{RANDOM}(p, r) \)
2 exchange \( A[r] \) with \( A[i] \)
3 \textbf{return} \ PARTITION\((A, p, r)\)

Call \textbf{RANDOMIZED-PARTITION}(A, p, r) \ instead \of \ PARTITION(A, p, r)
Bounding Runtime

Lemma

Let $X$ be the number of comparisons performed in line 4 of `PARTITION` over the entire execution of `QUICKSORT` on an $n$-element array. Then the running time of `QUICKSORT` is $O(n + X)$

Proof

- There are at most $n$ calls to `PARTITION`
- Each call takes $O(1)$ time to setup
- The loop in lines 3 – 6 is the only place where we make a comparison
Renaming elements of $A$ as $z_1, z_2, \ldots, z_n$, with $z_i$'s sorted

Define $Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$ to be the set of elements between $z_i$ and $z_j$, inclusive.

Let $X_{ij}$ be an indicator variable:

$$X_{ij} = I\{z_i \text{ is compared to } z_j\}$$

Therefore

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

\[
\mathbb{E} [X] = \mathbb{E} \left[ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} \right] \\
= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E} [X_{ij}] \\
= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr \{ z_i \text{ is compared to } z_j \} .
\]
Observation - 1

- If \( z_i < x < z_j \) is chosen as a pivot then \( z_i \) and \( z_j \) are never compared.
Observation - II

- If $z_i$ is chosen as a pivot before any other item in $Z_{ij}$, then $z_i$ is compared with each item in $Z_{ij}$
- Same holds for $z_j$
Probability of Comparison

\[ \Pr \{ z_i \text{ is compared to } z_j \} = \Pr \{ z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij} \} \]
\[ = \Pr \{ z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij} \} + \Pr \{ z_j \text{ or } z_i \text{ is first pivot chosen from } Z_{ij} \} \]
\[ = \frac{1}{j - i + 1} + \frac{1}{j - i + 1} \]
\[ = \frac{2}{j - i + 1} \]
**Expectations**

\[
\mathbb{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \\
\leq \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \\
\sum_{i=1}^{n-1} O(\log n) \\
= O(n \log n)
\]
Questions?