CMPT307: Quicksort

Week 5-1

Xian Qiu

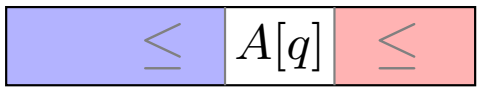
Simon Fraser University

xianq@sfu.ca

Quicksort

- ▷ divide: rearrange $A[p..r]$ into $A[p..q-1]$ and $A[q+1..r]$ s.t.

$$A[i] \le A[q] \le A[j], \quad \forall p \le i \le q-1 \text{ and } q+1 \le j \le r$$



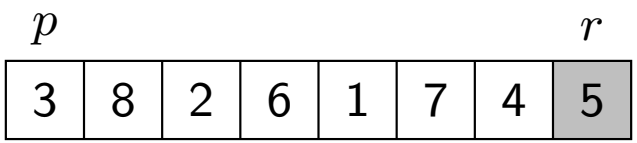
- ▷ conquer: sort $A[p..q-1]$ and $A[q+1..r]$ recursively
- ▷ combine: no need (subarrays are already sorted)

QUICKSORT(A, p, r)

```
1 if p < r then
2     q = PARTITION(A, p, r);                    // rearrangement
3     QUICKSORT(A, p, q − 1);
4     QUICKSORT(A, q + 1, r);
```

Partitioning

- select $A[r]$ as pivot
- if $A[j] < A[r]$ for $p \leq j < r$, color it blue else color it red
- make switches if red and blue appear alternatively



Pseudocode

PARTITION(A, p, r)

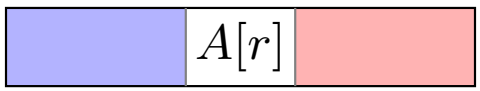
```
1 x = A[r];                                   // select A[r] as pivot
2 i = p − 1;
3 for j = p to r − 1 do
4     if A[j] ≤ x then
5         i = i + 1;
6         exchange A[i] with A[j];
7 exchange A[i + 1] with A[r];
8 return i + 1;
```

Performance

worst-case: $A[r] = \max_i A[i]$

$$T(n) = T(n-1) + T(0) + \Theta(n) \Rightarrow T(n) = \Theta(n^2)$$

best-case: $A[r]$ is median

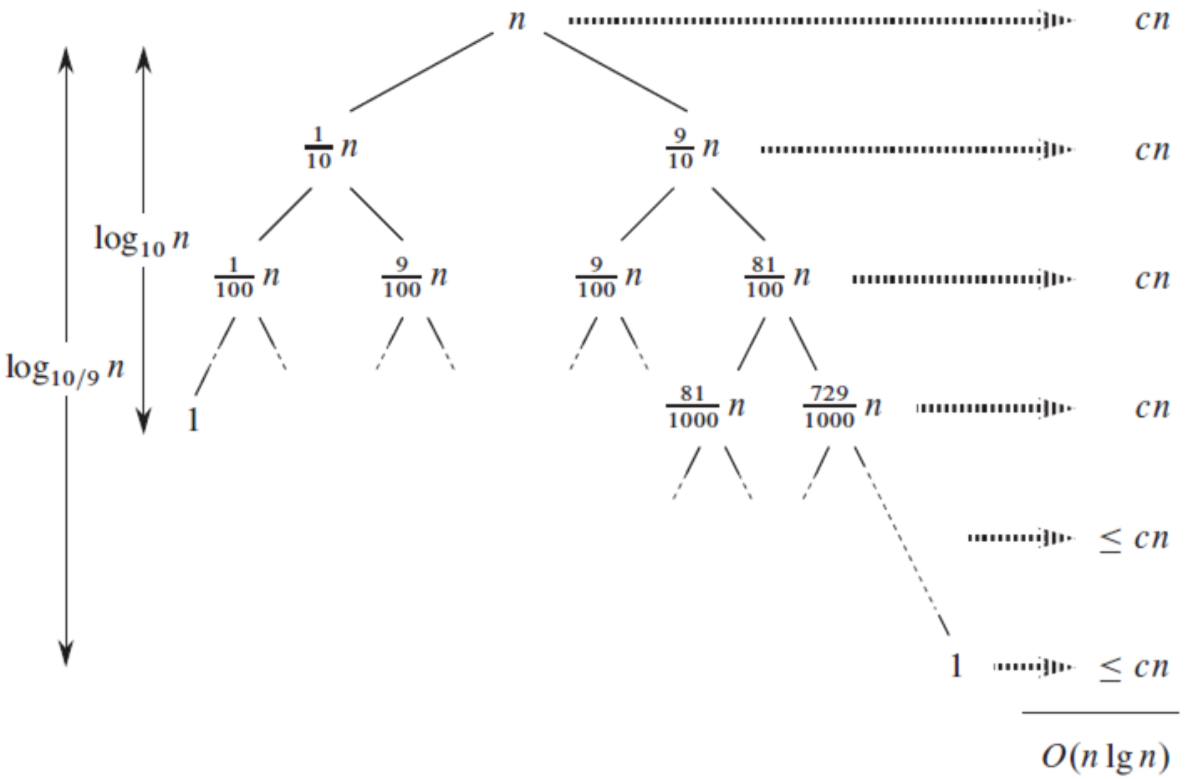


$$T(n) = 2T(n/2) + \Theta(n) \Rightarrow T(n) = \Theta(n \log n)$$

Performance

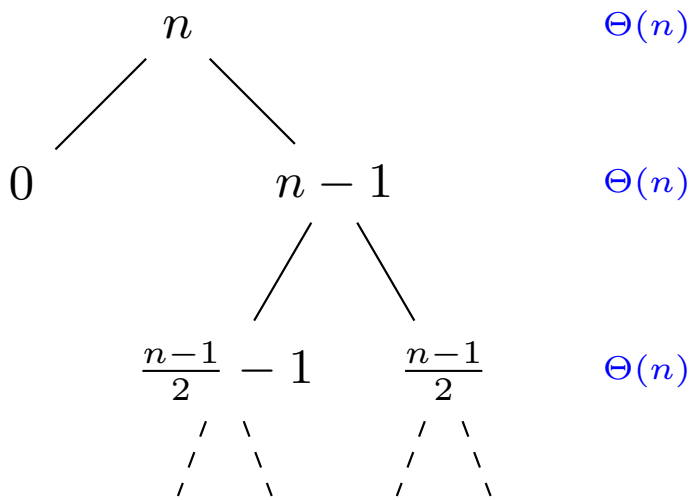
what if slightly better than worst-case? say

$$T(n) = T(9n/10) + T(n/10) + cn$$



Average-Case

- assume the input is a uniform random permutation
- PARTITION produces a mix of "worst" and "best" splits



$h \leq 2 \log n$

- intuition: $O(n \log n)$
- what if the assumption does not hold?

Randomized Partition

random sampling

- ▷ randomly select $A[i]$ from $A[p..r]$ as pivot
- ▷ exchange $A[r]$ with $A[i]$ and apply PARTITION;

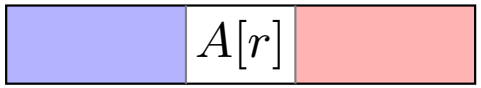
RANDOMIZED-PARTITION(A, p, r)

```
1 i = RANDOM(p, r);
2 exchange A[i] with A[r];
3 return PARTITION(A, p, r);
```

- ▷ use RANDOMIZED-PARTITION in QUICKSORT

Analysis

- each pivot appears exactly once



  # calls to PARTITION $\leq n$

- define $X :=$ # comparisons to pivots

$$T(n) = O(n + X) = O(X)$$

- let z_i be the ith smallest element and define

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

  assume all values are distinct

- how to express X as a function of X_{ij}?

Analysis

when does it compare z_i and z_j?

- z_i or z_j is pivot (only apears in at most one loop)
- the pair z_i, z_j is compared at most once

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

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

$$\Pr\{X_{ij} = 1\} = ?$$

Analysis

- define $Z_{ij} = \{z_i, \ldots, z_j\}$ for $i < j$ assume all distinct values
- if x_k with $i < k < j$ is chosen as pivot, then z_i and z_j will not be compared at any subsequent time

$$\begin{aligned}
&\Pr\{z_i \text{ is compared to } z_j\} \\
&= \Pr\{z_i \text{ or } z_j \text{ is the first pivot chosen from } Z_{ij}\} = \frac{2}{j-i+1}
\end{aligned}$$

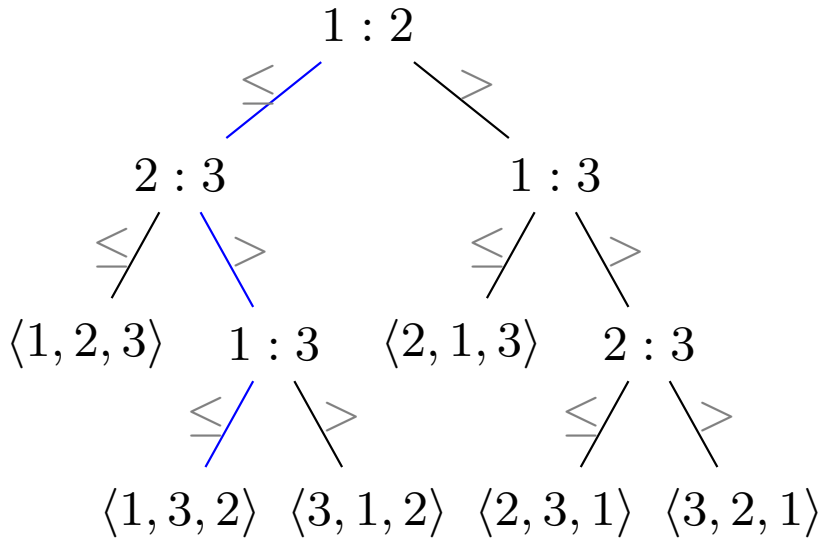
$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \\
&< \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} = \sum_{i=1}^{n-1} O(\log n) = O(n \log n)
\end{aligned}$$

Remarks on Quicksort

- though $\Theta(n^2)$ in worst case, assuming distinct elements, it is $O(n \log n)$ in expectation
- efficient in practice and the constant hidden in big O notation is small
- in place algorithm
- can we break the barrier $O(n \log n)$?

Decision Tree for Sorting

comparison sort: sorting algorithm based on comparison



- ▷ #leaves $= |\{\text{possible inputs}\}| = \#\text{permutations} = n!$
- ▷ any path from root to leaf corresponds to an execution of a comparison sorting algorithm
- ▷ worst case running time = height of decision tree

Lower Bound

Theorem

Comparison sort requires $\Omega(n \log n)$ comparisons in the worst case.

consider a decision tree of height h with l leaves

- $l \leq 2^h$ for any binary tree
- algorithm must solve all permutations, thus $l \geq n!$

$$n! \leq l \leq 2^h \Rightarrow h \geq \log(n!) = \Omega(n \log n)$$

Stirling formula: $n! = \sqrt{2\pi n}\left(\frac{n}{3}\right)^n (1 + \Theta(1/n))$