6

Tuesday, October 15, 2019 2:10 PM

Randomized Quicksort

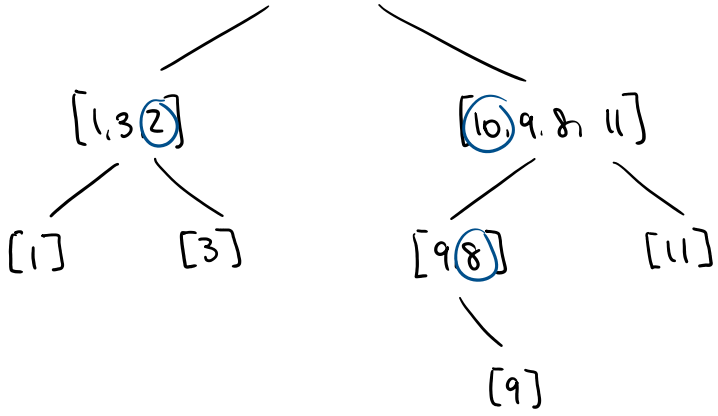
$A[1 \dots n]$ array of integers (distinct)

QuickSort ($A[1 \dots n]$)

- Pick pivot p from $\{1, \dots, n\}$ uniformly at random
- $A_< =$ array of $A[i]$ such that $A[i] < A[p]$
- $A_> =$ array of $A[i]$ such that $A[i] > A[p]$
- QuickSort ($A_<$)
- QuickSort ($A_>$)
- $A = [A_<, A, A_>]$

Thm: Randomized quicksort has worst-case expected running time $\Theta(\log n)$.

Example: $A = [10, 1, 3, 9, 2, 7, 8, 11]$



This is a BST.

Question: How many times can $A[i]$ and $A[j]$ be compared?

min = 0, max = 1

- Compared only with the pivot element.
- Once in different subtrees, they are never compared again.

$A[i] < \text{pivot} < A[j]$

Let r_i = index of element of rank i in A. So:

$A[r_1] < A[r_2] < \dots < A[r_n]$, name these $z_1, \dots, z_n$.

Let $X_{ij} = \begin{cases} 1 & \text{if } A[r_i] \text{ and } A[r_j] \text{ were ever compared} \\ 0 & \text{otherwise} \end{cases}$

Then:

$$\text{total number of comparisons} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

$$\mathbb{E}[\#\text{ comps}] = \mathbb{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} \right]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[X_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P(X_{ij} = 1)$$

Suppose that in some recursive call, the array (in sorted order) is:

$$[z_k, \dots, z_i, \dots, z_j, \dots z_\ell]$$

then, if we pick pivot p:

- If $A[p] < z_i$ or $A[p] > z_j$:
  then X_{ij} is undecided
- If $A[p] = z_i$ or $A[p] = z_j$
  then $X_{ij} = 1$
- Otherwise, $z_i < A[p] < z_j$, $X_{ij} = 0$.

Let D_{ij} = node in the recursion tree at which X_{ij} gets decided.

(... $z_i < A[p] < z_j$)

Let D_{ij} = node in the recursion tree at which X_{ij} gets decided.
(where $z_i \le A[p] \le z_j$)

$$P(X_{ij}=1) = \sum_{u \in \text{rec tree}} P(X_{ij}=1 \text{ and } D_{ij}=u)$$

$$= \sum_u P(X_{ij}=1 \mid D_{ij}=u) \cdot P(D_{ij}=u)$$

$$= \sum_u \frac{2}{j-i+1} \cdot P(D_{ij}=u)$$

$$= \frac{2}{j-i+1} \sum_u P(D_{ij}=u)$$

$$= \boxed{\frac{2}{j-i+1}}$$

Principle of Deferred Decision

→ Must be 1 since one node needs to be the deciding node

Then:

$$\mathbb{E}[\#\text{ comps}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

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

change of var $k = j-i+1$

$$\le 2 \cdot \sum_{i=1}^{n-1} \boxed{\sum_{k=1}^{n} \frac{1}{k}}$$

→ $\Theta(\log n)$

$$= \Theta(n \cdot \log n)$$

■

Amortized Analysis

If using DS in an algorithm, don't care about complexity for each operation, but care about the total running time. (Offline algorithms)

Have data structure D. $\sigma = (\sigma_1, \ldots, \sigma_m)$ operations on D.

$$D_0 \xrightarrow{\sigma_1} D_1 \xrightarrow{\sigma_2} D_2 \longrightarrow \cdots \xrightarrow{\sigma_m} D_m$$

$D_0 \xrightarrow{\sigma_1} D_1 \xrightarrow{\sigma_2} D_2 \longrightarrow \dots \xrightarrow{\sigma_m} D_m$

↖ state of the DS.

Let $t(\sigma_i)$ = time to do operation σ_i, and $t(\sigma) = \sum_i t(\sigma_i)$

Let (seq. complexity):

$$C(m) = \max\{t(\sigma) : \sigma \text{ sequence of } m \text{ ops}\}$$

(amortized complexity)

$$A(m) = \frac{C(m)}{m}$$ average time for each operation

Note: For upper bound: for all σ.
lower bound: show a "bad" σ.

Example: (Binary Counter) Count up from 0 to 2^{k-1} in binary.

DS: $A[0 \dots k-1]$ represents $\sum_{i=0}^{k-1} A[i] \cdot 2^i$ and $A[i] \in \{0, 1\}$.

ops: Increment(A):

```
i = 0
while i < k and A[i] == 1:
    A[i] = 0
    i = i+1
if i < k:   A[i] = 1
```

Worst case: $\Theta(k)$

Amortized complexity: (starting from $a = 0$) $A(m) = \Theta(1)$

Counting bit flips:

Claim: In a sequence of m Increments, the number of bit flips is bounded by $2m$.

Observations:

1) A[1] (LSB) flipped with every operation.
2) A[2] flipped with every second operation

1) A[1] (LSB) flipped with every operation.

2) A[2] flipped with every second operation

3) A[3] flipped with every 4 operations

Argument: A[i] flipped when A[i] if A[i-1], A[i-2], ..., A[0] are 1.

We can count flips for each of the bits.

A[i] flipped $\left\lfloor \frac{m}{2^i} \right\rfloor$ times in m ops.

$$\text{Total flips} \leq \sum_{i=0}^{k-1} \left\lfloor \frac{m}{2^i} \right\rfloor$$

$$\leq \sum_{i=0}^{\infty} \frac{m}{2^i}$$

$$= 2m$$