CS473-Algorithms I

Lecture 6-b

Randomized QuickSort

Randomized Quicksort

- Average-case assumption:
  - all permutations are equally likely
  - cannot always expect to hold

- Alternative to assuming a distribution: Impose a distribution
  - Partition around a random pivot

Randomized Quicksort

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

- Running time is independent of input ordering
- No specific input causes worst-case behavior
- Worst case determined only by output of random number generator

Randomized Quicksort

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

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

```
exchange A[r] ↔ A[s]
return L-PARTITION(A, p, r)
for Lomuto's partitioning
```

- Permuting whole array also works well on the average
  - more difficult to analyze

Formal Average - Case Analysis

- Assume all elements in A[$p \ldots r$] are distinct
- $n = r - p + 1$
- $rank(x) = |\{A[i]: p \leq i \leq r \text{ and } A[i] \leq x\}|$
- "exchange $A[p] \leftrightarrow x = A[s]$" ($x \in A[p \ldots r]$ random pivot)

$\Rightarrow P(rank(x) = i) = 1/n$, for $i = 1, 2, \ldots, n$

Likelihood of Various Outcomes of Hoare's Partitioning Algorithm

x = pivot

- $rank(x) = 1$:

  $k = 1$ with $i_1 = j_1 = p \Rightarrow L_1 = \{A[p] = x\}$

  $\Rightarrow |L| = 1$

- $rank(x) > 1$: $\Rightarrow k > 1$
  - *iteration* 1: $i_1 = p,\ p < j_1 \leq r \Rightarrow \quad A[p] \leftrightarrow x = A[j_1]$

    $\Rightarrow$ pivot x stays in the right region
  - *termination*: $L_k = \{A[i]:\ p \leq i \leq r$ and $A[i] < x\}$

  $\Rightarrow |L| = rank(x) - 1$

Various Outcomes

- $rank(x) = 1 : \Rightarrow |L| = 1$
- $rank(x) > 1 : \Rightarrow |L| = rank(x) - 1$

x = pivot

- $P(|L|=1) = P(rank(x) = 1) + P(rank(x) = 2)$

$$= 1/n + 1/n = 2/n$$

- $P(|L|=i) = P(rank(x) = i+1)$

$$= 1/n \qquad \text{for } i = 2, \ldots, n-1$$

Average - Case Analysis: Recurrence

		$rank(x)$
$T(n) =$	$\frac{1}{n}(T(1)+T(n-1))$	1
$+$	$\frac{1}{n}(T(1)+T(n-1))$	2
$+$	$\frac{1}{n}(T(2)+T(n-2))$	3
$\vdots$	$\vdots$	$\vdots$
$+$	$\frac{1}{n}(T(\mathrm{i})+T(n-\mathrm{i}))$	$i+1$
$\vdots$	$\vdots$	$\vdots$
$+$	$\frac{1}{n}(T(\mathrm{n}-1)+T(1))$	n
$+$	$\Theta(n)$	

x = pivot

Recurrence

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

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

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

- 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. : $T(k) \leq ak\lg k + b \Rightarrow k<n$, for some constants $a > 0$ and $b \geq 0$

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

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

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

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

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

Tight bound for $\sum k \lg k$

- Bounding the terms

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

This bound is not strong enough because

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

$$= \quad 2an \lg n + 2b + \Theta(n)$$

Tight bound for $\sum k\lg k$

- Splitting summations: ignore ceilings for simplicity

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

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

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

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

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

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

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

$$\sum_{k=1}^{n-1} k \lg k \le \frac{1}{2} n^2 \lg n - \frac{1}{8} n^2 \quad \mathit{for} \lg n \ge 1/2 \Rightarrow n \ge \sqrt{2}$$

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

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

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

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

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

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