CS 457, Fall 2019

Drexel University, Department of Computer Science

Lecture 7

Worst case linear time selection

Select(A,p,r,i)

1. Divide **A** into $n/5$ groups of size 5.
2. Find the median of each group of 5 by brute force, and store them in a set $\mathbf{A'}$ of size $n/5$.
3. Recursively use **Select**$(\mathbf{A'},\ \mathbf{1},\ \boldsymbol{n/5}, \boldsymbol{n/10})$ to find the median $\mathbf{x}$ of $\boldsymbol{n/5}$ medians.
4. Partition elements of **A** around $\mathbf{x}$. Let $\boldsymbol{k}$ be the order of $\mathbf{x}$ found in the partitioning.
5. if $\boldsymbol{i = k}$
6. return $\mathbf{x}$
7. else if $\boldsymbol{i < k}$
8. **Select**$(\mathbf{A},\ \mathbf{p},\ \mathbf{q-1},\ \boldsymbol{i})$
9. else
10. **Select**$(\mathbf{A}, \mathbf{q+1}, \mathbf{r}, \boldsymbol{i-k})$

Steps 1–2: $O(n)$

Step 3: $T\left(\left\lceil \frac{n}{5} \right\rceil\right)$

Step 4: $O(n)$

Steps 5–10: $T\left(\frac{7n}{10} + 6\right)$

$$T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + O(n)$$

Worst case linear time selection

Guess that $T(n) = O(n)$ when $T(n) \leq T\left(\left\lceil\frac{n}{5}\right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + O(n)$

Assume $T(n') \leq cn'$ for some constant c and every $n' < n$, and

show that this implies $T(n) \leq cn$ (for the same constant c)

Upon substitution, we get $T(n) \leq c\left\lceil\frac{n}{5}\right\rceil + c\left(\frac{7n}{10} + 6\right) + O(n)$

For $n > n_0$ this is at most $c\left(\frac{n}{5} + 1\right) + c\left(\frac{7n}{10} + 6\right) + an$

Therefore, $T(n) \leq cn - \left(\frac{cn}{10} - 7c - an\right)$

Worst case linear time selection

Guess that $T(n) = O(n)$ when $T(n) \leq T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + O(n)$

Assume $T(n') \leq cn'$ for some constant c and every $n' < n$, and

show that this implies $T(n) \leq cn$ (for the same constant c)

Upon substitution, we get $T(n) \leq c\left\lceil \frac{n}{5} \right\rceil + c\left(\frac{7n}{10} + 6\right) + O(n)$

For $n > n_0$ this is at most $c\left(\frac{n}{5} + 1\right) + c\left(\frac{7n}{10} + \ldots\right.$

for $c \geq 20a$
and $n \geq 140.$

Therefore, $T(n) \leq cn - \left(\frac{cn}{10} - 7c - an\right) \leq cn$

Simple Selection Algorithm

Select(A, p, r, i)

if $p == r$
    return $A[p]$
q = Partition(A, p, r)
$k = q - p + 1$
if $i == k$
    return $A[q]$
else if $i \leq k$
    Select$(A,\ p,\ q-1,\ i)$
else
    Select$(A,\ q+1,\ r,\ i-k)$

Partition (A, p, r)

$x = A[r]$
$i = p - 1$
for $j = p$ to $r - 1$
    if $A[j] \leq x$
        $i = i + 1$
        exchange $A[i]$ with $A[j]$
exchange $A[i+1]$ with $A[r]$
return $i + 1$

Randomized Selection Algorithm

Randomized-Select(A, p, r, i)

```
if p == r
    return A[p]
q = Randomized-Partition(A, p, r)
k = q − p + 1
if i == k
    return A[q]
else if i ≤ k
    Randomized-Select(A, p, q − 1, i)
else
    Randomized-Select(A, q + 1, r, i − k)
```

Randomized-Partition (A, p, r)

```
i = Random(p, r)
Exchange A[r] with A[i]
return Partition(A, p, r)
```

Randomized Selection Algorithm

Randomized-Select(A, p, r, i)

if $p == r$
    return $A[p]$
q = Randomized-Partition(A, p, r)
$k = q - p + 1$
if $i == k$
    return $A[q]$
else if $i \leq k$
    Randomized-Select$(A,\ p,\ q-1,\ i)$
else
    Randomized-Select$(A,\ q+1,\ r,\ i-k)$

- Worst-case running time?
  - a) $O(n^2)$
  - b) $O(n \log n)$
  - c) $O(n)$
- Expected running time?
  - a) $O(n^2)$
  - b) $O(n \log n)$
  - c) $O(n)$

Randomized Selection Algorithm

Randomized-Select(A, p, r, i)

if $p == r$
    return $A[p]$
q = Randomized-Partition(A, p, r)
$k = q - p + 1$
if $i == k$
    return $A[q]$
else if $i \leq k$
    Randomized-Select$(A,\ p,\ q-1,\ i)$
else
    Randomized-Select$(A,\ q+1,\ r,\ i-k)$

Randomized-Partition (A, p, r)

i = Random(p, r)
Exchange $A[r]$ with $A[i]$
return Partition(A, p, r)

Running Time

- Indicator variable $\mathbb{I}\{E\}$ is 1 if event E occurs and 0 o/w (see page 118)
- Consider an array $A[p, \ldots, r]$ with n elements
- If $X_k = \mathbb{I}\{\text{the subarray } A[p, \ldots, q] \text{ has exactly } k \text{ elements}\}$, then

$$T(n) \leq \sum_{k=1}^{n} X_k \left(T(\max\{k-1, n-k\}) + O(n)\right)$$

$$= \sum_{k=1}^{n} X_k \left(T(\max\{k-1, n-k\})\right) + \sum_{k=1}^{n} X_k \, O(n)$$

$$= \sum_{k=1}^{n} X_k \left(T(\max\{k-1, n-k\})\right) + O(n)$$

Expected Running Time

- $X_k = \mathbb{I}\{\text{the subarray } A[p,\dots,q] \text{ has exactly } k \text{ elements}\}$. Since $A[p,\dots,r]$ has n elements and q is chosen uniformly at random, we have $\mathbb{E}[X_k] = \frac{1}{n}$, so

$$\mathbb{E}[T(n)] \le \mathbb{E}\left[\sum_{k=1}^{n} X_k\,(T(\max\{k-1, n-k\})) + O(n)\right]$$

$$= \sum_{k=1}^{n} \mathbb{E}[X_k(T(\max\{k-1, n-k\}))] + O(n)$$

$$= \sum_{k=1}^{n} \mathbb{E}[X_k]\,\mathbb{E}[T(\max\{k-1, n-k\})] + O(n)$$

$$= \sum_{k=1}^{n} \frac{1}{n}\,\mathbb{E}[T(\max\{k-1, n-k\})] + O(n)$$

Expected Running Time

Thus, $\mathbb{E}[T(n)] \le \sum_{k=1}^{n} \frac{1}{n}\, \mathbb{E}[T(\max\{k-1, n-k\})] + O(n)$,

and $\max\{k-1, n-k\}) = \begin{cases} k-1 \text{ if } k > \lceil n/2 \rceil \\ n-k \text{ if } k \le \lceil n/2 \rceil \end{cases}$

This is a recurrence that we can now solve

So, $\mathbb{E}[T(n)] \le \frac{2}{n} \sum_{k=\lceil \frac{n}{2} \rceil}^{n-1} \mathbb{E}[T(k)] + O(n)$

Expected Running Time

We use substitution in order to solve: $\mathbb{E}[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} \mathbb{E}[T(k)] + O(n)$

We guess that $\mathbb{E}[T(n)] = O(n)$

Hence, we assume $\mathbb{E}[T(n')] \leq cn'$ for some constant c and every $n' < n$,

and we show that $\mathbb{E}[T(n)] \leq cn$ for that same constant c. Upon substitution:

$$\mathbb{E}[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck + an \quad \leq \frac{2c}{n}\left(\frac{n^2 - n}{2} - \frac{n^2/4 - 3n/2 + 2}{2}\right) + an$$

which leads to $\mathbb{E}[T(n)] \leq cn - \left(\frac{cn}{4} - \frac{c}{2} - an\right)$

Expected Running Time

We use substitution in order to solve: $\mathbb{E}[T(n)] \le \frac{2}{n} \sum_{k=\left\lfloor \frac{n}{2} \right\rfloor}^{n-1} \mathbb{E}[T(k)] + O(n)$

We guess that $\mathbb{E}[T(n)] = O(n)$

Hence, we assume $\mathbb{E}[T(n')] \le cn'$ for some constant c and every $n' < n$,

and we show that $\mathbb{E}[T(n)] \le cn$ for that same constant c. Upon substitution:

$$\mathbb{E}[T(n)] \le \frac{2}{n} \sum_{k=\left\lfloor \frac{n}{2} \right\rfloor}^{n-1} ck + an \le \frac{2c}{n}\left(\frac{n^2-n}{2} - \cdots\right) + an$$

for $c > 4a$,

and $n \ge \frac{2c}{c-4a}$

which leads to $\mathbb{E}[T(n)] \le cn - \left(\frac{cn}{4} - \frac{c}{2} - an\right) \le cn$

Today's Lecture

- More probabilistic analysis and randomized algorithms
  - Randomized Quicksort
  - Hash tables

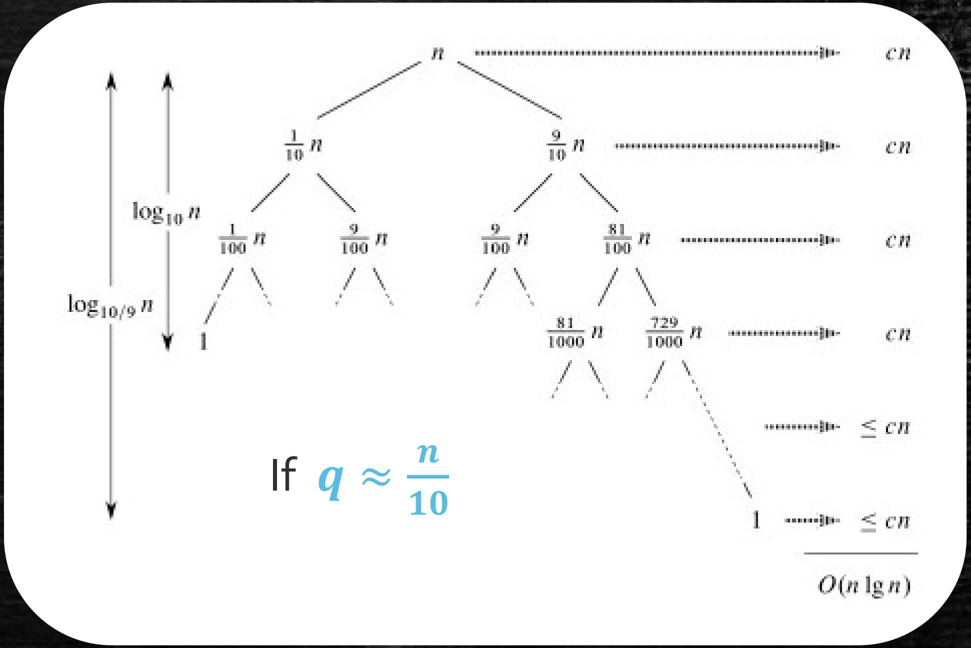
Quicksort (Running Time)

QUICKSORT (*A*, *p*, *r*)

1. **if** $p < r$ // Check for base case
2. q = PARTITION(A, *p*, *r*) // Divide step
3. QUICKSORT (A, *p*, q - 1) // Conquer step
4. QUICKSORT (A, q + 1, *r*) // Conquer step

$$\boldsymbol{T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(q) + T(n-q-1) + \Theta(n) & \text{otherwise} \end{cases}}$$

- $T(n) \leq \max_{0 \leq q \leq n-1} (T(q) + T(n-q-1)) + \Theta(n)$
- This leads to $O(n^2)$ in the worst case
- But, if $q = cn$ for some constant c, it is $O(n \log n)$



Quicksort (Running Time)

How many times is Partition called overall?

QUICKSORT (A, p, r)

```
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT (A, p, q - 1)
    QUICKSORT (A, q + 1, r)
```

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
```

What is the worst case value of X?

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 $O(n+X)$.

Quicksort (Running Time)

RANDOMIZED-QUICKSORT (A, p, r)

```
if p < r
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT (A, p, q - 1)
    RANDOMIZED-QUICKSORT (A, q + 1, r)
```

RANDOMIZED-PARTITION (A, p, r)

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

What is the expected value of X if we choose the pivot element randomly?

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 $O(n + X)$.

Randomized Quicksort (Running Time)

- Denote the sorted elements of the array by $z_1, z_2, \dots, z_n$
- Let $X_{ij} = I\{z_i \text{ is compared to } z_j\}$
- Then, $X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$ and $E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr\{z_i \text{ is compared to } z_j\}$

Check out Appendix C

- Let $Z_{ij} = \{z_i, z_{i+1}, \dots, z_j\}$
- $Pr\{z_i \text{ is compared to } z_j\} = Pr\{z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij}\} = \frac{2}{j-i+1}$
- So, $E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} < \sum_{i=1}^{n-1} \sum_{k=1}^{n-1} \frac{2}{k} = \sum_{i=1}^{n-1} O(\log n) = O(n \log n)$