Selection algorithm

Tufts University

Warm-up question

Selection: given array and $k \leq n$ find the k-th smallest element in array

Example: for $k = 6$ return 9

2	20	22	18	6	18	5	7	8	9

Design a fast algorithm to solve this problem

Previously ...

Three tools to solve recurrences:

- Recursion trees
  - ☺ Very intuitive. Just draw!
  - ☹ Hard to have matching upper and lower bounds
- Substitution
  - ☺ Math heavy. Most powerful
  - ☹ Needs a hunch
- Master method
  - ☺ Good! Pretty! Fast!
  - ☹ Cannot always use

Master theorem review for $T(n) = aT(\frac{n}{b}) + f(n)$

Three cases:

Leaves dominate $f(n)$ is **strictly smaller** than $n^{\log_b a}$

All levels equal $f(n)$ is **almost equal** to $n^{\log_b a}$

Top dominates $f(n)$ is **strictly larger** than $n^{\log_b a}$

Note: just intuition. See previous lecture for formal definitions

Definition

$\textsc{Selection}$(A,n,k): find k-th smallest in array A of n numbers

- $\textsc{Selection}(A, n, 1)$=min$\{A[i]\}$
- $\textsc{Selection}(A, n, n)$=max$\{A[i]\}$
- $\textsc{Selection}(A, n, \frac{n}{2})$=$\textsc{Median}\{A[i]\}$

Simple algorithm? Sort, return $A[k]$

Today's goal: linear time algorithm for any k

Road to a linear time algorithm

Algorithm by Blum, Floyd, Pratt, Rivest, Tarjan (1973)

7 Key steps:

- PARTITION
- Algorithm overview
- Correctness
- Picking an index
- Recursive formula
- Crop Lemma
- Runtime

Step 0: Cheat of the day

Lemma

For any $n \geq 50$ it holds that $3\lfloor \frac{\lfloor \frac{N}{5} \rfloor}{2} \rfloor \geq \frac{n}{4}$

Step 1: $\textsc{Partition}(A, n, k)$

Given an array and index k rearrange it so that:

- All numbers smaller (or equal) than $A[k]$ go **first**
- $A[k]$ in **middle**
- Numbers larger than $A[k]$ go **last**

Return final position of $A[k]$

2	20	22	18	6	18	5	7	8	9

$\Downarrow$

smaller					$A[k]$	larger			
2	8	7	5	6	9	22	18	20	18

Return 6 (because $A[6] = 9$)

Partition, part 2

Arrange three intervals in A:

- $A[1]$ contains original $A[k]$
- $\mathbb{s}$ elements smaller than $A[k]$
- ℓ elements larger than $A[k]$
- Rest are unexplored

$A[k]$	smaller	larger	unknown

Initial values of $\mathbb{s}$ and ℓ? $\mathbb{s} = \ell = 0$

First unexplored spot? $A[\mathbb{s} + \ell + 2]$

If $A[k] < A[\mathbb{s} + \ell + 1] \Rightarrow \ell \leftarrow \ell + 1$

Else Swap $A[\mathbb{s} + 2]$ and $A[\mathbb{s} + \ell + 2]$; $s \leftarrow s + 1$

Practice

2	20	22	18	6	18	5	7	8	9

Run PARTITION step-by-step on previous instance

- Any initialization?
- Show how s and ℓ change along time
- Any final steps?
- How to return position of $A[k]$?

Give the pseudocode of the algorithm

Step 2: Algorithm description

$\textsc{Selection}$(A,n,k)

if $n \leq 5$, $k = 1$ or $k = n$ solve by brute force

$\texttt{i} \leftarrow$ Magically pick an index(A, n)

$\texttt{pos} \leftarrow \textsc{Partition}(A, n, i)$

if $\texttt{pos} = k$ return $A[pos]$

if $\texttt{pos} > k$ return $\textsc{Selection}(A[1, \texttt{pos} - 1], \texttt{pos} - 1, k)$

else return $\textsc{Selection}(A[\texttt{pos} + 1, n], n - \texttt{pos}, k - \texttt{pos})$

Example: $\textsc{Selection}(A, 10, 6)$

2	20	22	18	6	18	5	7	8	9

If base case $\Rightarrow$ brute force

$\mathrm{v} \leftarrow \textsc{PickIndex}(A, n)$

$\mathrm{pos} = \textsc{Partition}(A, n, v)$

If $\mathrm{pos} = k$ done

Recurse if $(\mathrm{pos} > k)$ or $(\mathrm{pos} < k)$

Step 3: Correctness

Lemma

Regardless of how we pick index, the algorithm is always correct

Proof.

By induction

Base case: we solve by brute force

Induction step three cases:

- If $\mathrm{pos} = k$ $\textsc{PickIndex}$ did a great job
- If $\mathrm{pos} > k$ We discard large values → Induction on $A[1, pos - 1]$
- If $\mathrm{pos} < k$ We discard small values and update rank → ...

$\square$

Step 4: Picking an index

PickIndex(A,n)

$B \leftarrow$ Empty array of size $n/5$

For $i = 0$ to $n/5$:

- $B[i] \leftarrow$ median of $A[5i], \ldots, A[5i+4]$

return $\textsc{Selection}(B, n/5, n/10)$ (actually, index in A)

Example:

3	15	12	11	4	2	20	22	18	6	18	5	7	8	9

Step 5: Runtime recurrence

$\textsc{Selection}$(A,n,k)

  if $n \leq 5$, $k = 1$ or $k = n$ solve by brute force

  $B \leftarrow$ Empty array of size $n/5$

  For $i = 1$ to $n/5$:

    $B[i] \leftarrow$ median of $A[5i+1], \ldots, A[5i+5]$

  $i \leftarrow$ index of $\textsc{Selection}(B, n/5, n/10)$ in A

  $\mathrm{pos} = \textsc{Partition}(A, n, i)$

  if $\mathrm{pos} = k$ return $A[pos]$

  if $\mathrm{pos} > k$ return $\textsc{Selection}(A[1, \mathrm{pos}-1], \mathrm{pos}-1, k)$

  else return $\textsc{Selection}(A[\mathrm{pos}+1, n], n-\mathrm{pos}, k-\mathrm{pos})$

Step 6: Crop Lemma

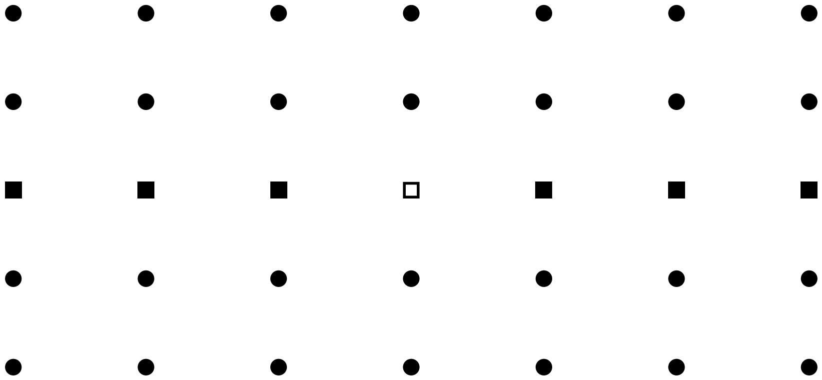
Lemma

If we pick index as described, then $\frac{n}{4} \le pos \le \frac{3n}{4}$

Implication? $T(n) \le \Theta(n) + T(\frac{n}{5}) + T(\frac{3n}{4})$

Proof: let's draw!

Proof of Crop Lemma



Step 7: Runtime

$$T(n) \leq \Theta(n) + T(\frac{n}{5}) + T(\frac{3n}{4})$$

Claim: $T(n) \leq cn$ for some $c > 0$
Proof by substitution:
Base Case $T(1) = O(1) \leq d$ ok as long as $c \geq d$
Induction Step

$$\begin{aligned} T(n) &\leq \Theta(n) + T(\frac{n}{5}) + T(\frac{3n}{4}) \\ &\leq d'n + c\frac{n}{5} + c\frac{3n}{4} \\ &= d'n + c\frac{19n}{20} \\ &\leq cn \qquad \text{ok as long as } d' \leq \frac{c}{20} \Leftrightarrow c \geq 20d' \end{aligned}$$

Glueing all steps together

Algorithm in 7 steps:

- PARTITION
- Algorithm overview
- Correctness
- Picking an index
- Recursive formula
- Crop Lemma
- Overall Runtime

Impressive!

Additional practice questions

- Make an array of 25 random numbers
- Run the Selection algorithm for $k = 12$
- Verify that each time pos satisfies the Crop Lemma
- Bonus: can you implement the algorithm? (**Future assignment!**)