

Medians and Order Statistics

Jonathan Backer
backer@cs.ubc.ca

Department of Computer Science
University of British Columbia



May 28, 2007

Introduction

Reading:

- ▶ CLRS: “Medians and Order Statistics” 9
- ▶ GT: “Sorting, Sets, and Selection” 4.7

Problem

How can we find the i^{th} smallest element of an unsorted array?
Sorting $\Theta(n \log n)$ will work, but we can do better.

Definition

The i^{th} order statistic of a set with n elements is the i^{th} smallest element of the set.

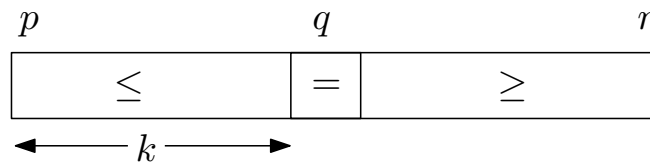
The min is the 1^{st} order statistic, the max is the n^{th} order statistic, and the median is the $\lfloor n/2 \rfloor^{th}$ order statistic.

5	8	13	21	34	55	89
2	4	8	16	32	64	128
				256		

QuickSort Review

```
Algorithm QuickSort( $A, p, r$ )  
  if  $p < r$  then  
     $q \leftarrow \text{Partition}(A, p, r)$ ;  
    QuickSort( $A, p, q - 1$ )  
    QuickSort( $A, q + 1, r$ )
```

Why sort both
partitions for selection?



Deterministic Divide-and-Conquer

```
Algorithm Select( $A, p, r, i$ )  
  if ( $p = r$ ) then  
    return  $A[p]$ ;  
   $q \leftarrow \text{Partition}(A, p, r)$ ;  
   $k \leftarrow q - p$ ;           // size of left side  
  if ( $i = k + 1$ ) then  
    return  $A[q]$ ;  
  else if ( $i \leq k$ ) then  
    return Select( $A, p, q - 1, i$ );  
  else  
    return Select( $A, q + 1, r, i - (k + 1)$ );
```

Deterministic Divide-and-Conquer (cont'd)

Worst-Case Complexity

- ▶ Look for the smallest.
- ▶ Always pivot around the largest element.
- ▶ Generates $\Theta(n^2)$ runtime.

Idea: Pick the pivot at random

Definition

An algorithm is **randomized** if its behaviour depends on the input and the values produced by a random number generator.

Randomized algorithms are nice when they are

- ▶ simple and
- ▶ have good average-case running times.

Randomized Divide-and-Conquer

Same worst case complexity because of bad guesses!

So randomly choose a good partition:

- ▶ Good if each side has at most $3n/4$ elements.
- ▶ $P[\text{partition is good}] \approx 1/2$ because every partition is equally likely.
- ▶ About two guesses to get a good partition, on average.
- ▶ So $O(n)$ time to get a good partition, on average.

Randomized Divide-and-Conquer (cont'd)

Worst case: Element is partition with at most $3n/4$ elements.

$$T(n) = \begin{cases} T(3n/4) + O(n) & \text{if } n \geq 0 \\ O(1) & \text{otherwise} \end{cases}$$

Case 3 Master Theorem: $T(n) \in O(n)$.

Not true average because we assumed element in large partition.

Deterministically Finding a Pivot

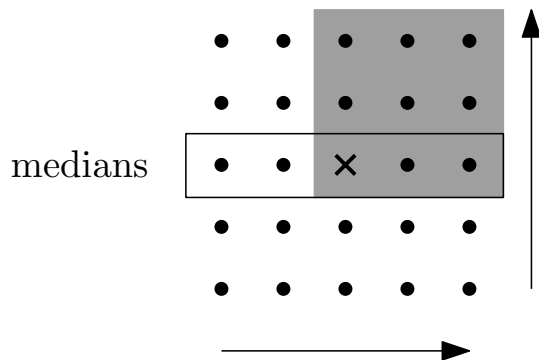
Must efficiently choose a guaranteed good pivot.

- ▶ Finding median too hard.
- ▶ Testing a constant number of elements too “local”.
- ▶ Elect samples to represent fixed-size subgroups.
- ▶ Recursively elect leader (pivot) from representatives.

Specifically,

- ▶ Each group of five elects representative (median).
- ▶ Recursively call selection to find the median of the medians.

Pivot quality



How many elements must be greater or equal to the pivot?

At least 3 for every representative greater than our pivot.

$$3\lfloor \lfloor n/5 \rfloor / 2 \rfloor \geq \text{pivot}$$

$$\begin{aligned}
 3\lfloor \lfloor n/5 \rfloor / 2 \rfloor &= 3\lfloor n/10 \rfloor \geq 3\lfloor n/10 - 1 \rfloor \\
 &= 3n/10 - 3 \\
 &= 6n/20 - 3 = (6n - 60)/20 \\
 &= (5n + n - 60)/20 \geq 5n/20, \text{ if } n \geq 60 \\
 &= n/4
 \end{aligned}$$

As n gets large, we eliminate at least one quarter of the elements.

Worst case complexity

Our worst case recurrence is

$$T(n) = \begin{cases} T(\lfloor n/5 \rfloor) + T(3n/4) + dn & \text{for } n > 60 \\ \Theta(1) & \text{otherwise} \end{cases}$$

We guess that $T(n) \leq cn$. Try the inductive step.

$$\begin{aligned}
 T(n) &\leq T(\lfloor n/5 \rfloor) + T(3n/4) + dn \\
 &\leq c\lfloor n/5 \rfloor + c \cdot 3n/4 + dn \text{ by inductive hypothesis} \\
 &\leq c \cdot n/5 + c \cdot 3n/4 + dn \\
 &= c \cdot 19n/20 + dn \\
 &\leq cn, \text{ if } n \geq 20d
 \end{aligned}$$

Only 60 base cases, so we can choose c large enough!