Algorithms

Recursion (final)
+ Backtracking

Recap

- Error on HW1! (Sorry.)

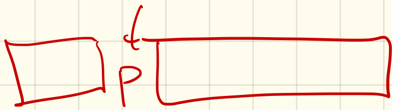
  #1: $E(n) = E(n-1)^2 - \underbrace{E(n-2)^2}$

- HW1 - due Friday
- HW2 - oral grading next Thursday + Friday
  (Sign up sheet available on Monday)

Linear Time Selection

First: Quicksearch

Idea: Modify quick sort, but don't recurse on both sides




```
QUICKSELECT(A[1..n], k):
  if n = 1
      return A[1]
  else
      Choose a pivot element A[p]
      r ← PARTITION(A[1..n], p)
      if k < r
          return QUICKSELECT(A[1..r − 1], k)
      else if k > r
          return QUICKSELECT(A[r + 1..n], k − r)
      else
          return A[r]
```

Figure 1.12. Quickselect, or one-armed quicksort

median of medians

Ex: Find 5th element (in sorted order) from:

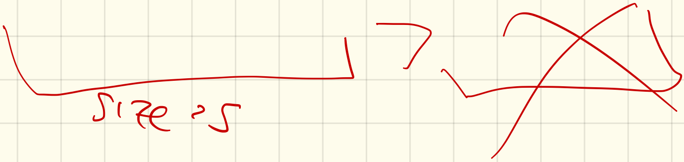
k

3 5 11 2 6 9 1 7 4 8 0

p

~~2 1 0~~ 3 5 11 6 9 7 4 8

p



Runtime:

Well, depends on pivot!

Ideal case:

divide in half

$$Q(n) = \overset{n}{\cancel{O(n)}} + Q\left(\frac{n}{2}\right)$$

$$= n + \frac{n}{2} + \frac{n}{4} + \cdots$$

$$= \sum \frac{n}{2^i} = O(n)$$

Worst case: pivot is 1st or last

$$Q(n) = O(n) + Q(n-1)$$

Bug last time!!

$$Q(n) = n + Q(n-1)$$

$$= n + (n-1) + Q(n-2)$$

$$\cdots = \sum_{i=1}^{n} i = O(n^2)$$

New algorithm:

```
MomSelect(A[1..n], k):
  if n ≤ 25   ⟨⟨or whatever⟩⟩
      use brute force
  else
      m ← ⌈n/5⌉
      for i ← 1 to m
          M[i] ← MedianOfFive(A[5i−4..5i])   ⟨⟨Brute force!⟩⟩
      mom ← MomSelect(M[1..m], ⌊m/2⌋)       ⟨⟨Recursion!⟩⟩

      r ← Partition(A[1..n], mom)

      if k < r
          return MomSelect(A[1..r−1], k)      ⟨⟨Recursion!⟩⟩
      else if k > r
          return MomSelect(A[r+1..n], k−r)    ⟨⟨Recursion!⟩⟩
      else
          return mom
```

"good pivot"

2 things:

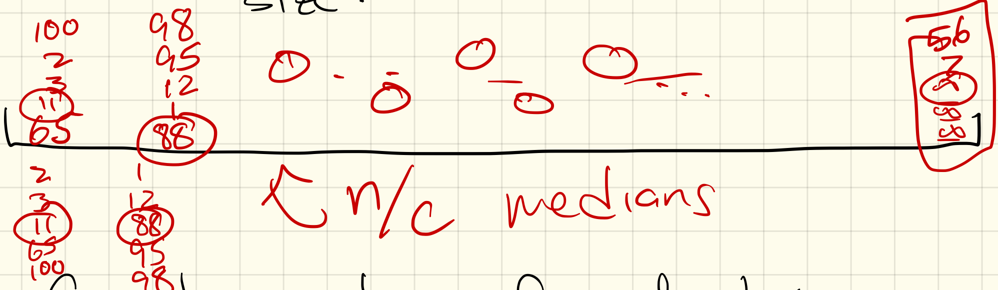
- Wierd 2nd recursion gives "good enough pivot"

- And linear time in the end

Improvement: Need a good enough pivot!

Median of medians: "MOM"

Divide into blocks that are $O(1)$ size:



Compute median of each by brute force

Runtime so far:

blocks of size c

n/c blocks

$$O(1) \times (\#\text{ blocks}) = O(n)$$

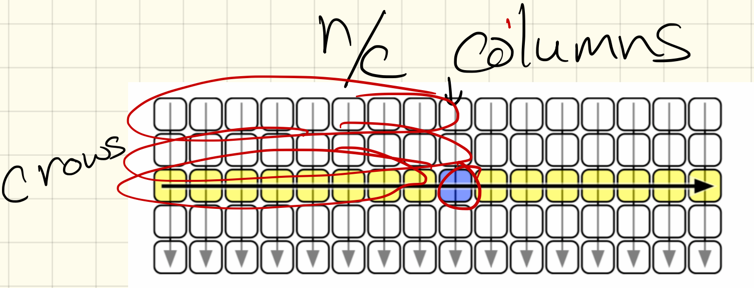
Then, recurse, & find median of medians.

Why does this work!?

Claim: MoM is a good pivot.

Why? Well, we've chopped the n elements into n/c groups of c.

Say $c = 5$:



n/c blocks, each holding c
($n/5$, each w/ 5)

The median of medians is $\geq$ at least
$\lceil \frac{c}{2} \rceil \cdot \left(\lceil \frac{n}{c} \rceil \cdot \frac{1}{2} \right)$ elements

With $n > 5$: $\lceil \frac{5}{2} \rceil \left(\lceil \frac{n}{5} \rceil \cdot \frac{1}{2} \right) \leq 3 \cdot \frac{n}{10} + 1$

Runtime: Let $M(n)$ be runtime on a list of size n

```
MomSelect(A[1..n], k):
  if n ≤ 25    ⟨⟨or whatever⟩⟩
      use brute force
  else
      m ← ⌈n/5⌉
      for i ← 1 to m
          M[i] ← MedianOfFive(A[5i−4..5i])    ⟨⟨Brute force!⟩⟩
      mom ← MomSelect(M[1..m], ⌊m/2⌋)         ⟨⟨Recursion!⟩⟩
      r ← Partition(A[1..n], mom)
      if k < r
          return MomSelect(A[1..r−1], k)       ⟨⟨Recursion!⟩⟩
      else if k > r
          return MomSelect(A[r+1..n], k−r)     ⟨⟨Recursion!⟩⟩
      else
          return mom
```

$O(n)$

$M\left(\frac{n}{5}\right)$

one call of size

Recurrence:

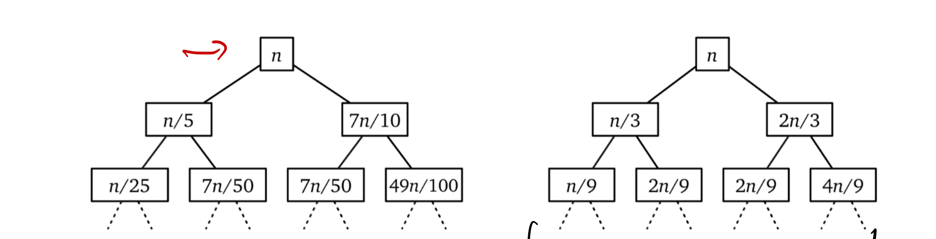
$$M(n) \leq M\left(\frac{n}{5}\right) + M\left(\frac{7n}{10}\right) + O(n)$$

finds good enough pivot

actual recursion on smaller list

Not a nice Master thm!

(Back to proof again...)



MoM with $c=5$ (good!)

Mom with $c=3$ (BAD!)

works, but $O(n \log n)$

$$M(n) \le n + \overbrace{\left(\frac{n}{5} + \frac{7n}{10}\right)}^{\frac{9}{10}\cdot n} + \underbrace{\left(\frac{n}{25} + \frac{7n}{50} + \frac{7n}{50} + \frac{49n}{100}\right)}_{\left(\frac{9}{10}\right)^2 \cdot n} + \cdots$$

$$= \sum_{i=0}^{\log_{10/9} n} \left(\frac{9}{10}\right)^i n \le O(n)$$

Another recursive strategy:
Backtracking (Chaper 2)

Idea: Build up a solution iteratively.

Setting: an algorithm needs to try multiple options.

Strategy: Make a recursive call for each possibility.

Downside: SLOW

Example: Subset Sum

Given a set X of positive integers and a target value t, is there a subset of X which sums to t?

Ex: $X = \{8, 6, 7, 3, 10, 5, 9\}$

$t = 15$

How would we solve?

Consider recursively:

$$X = \{8, 6, 7, 5, 3, 1, 9\}$$

Formalize this: recursion!

or base case?

Pseudocode:

```
SUBSETSUM(X[1..n], T):
  if T = 0
      return TRUE
  else if T < 0 or n = 0
      return FALSE
  else
      return (SUBSETSUM(X[1..n−1], T) ∨ SUBSETSUM(X[1..n−1], T − X[n]))
```

Runtime:

Correctness :

Next time:

Text segmentation