Read Chapter 10. We skip Chapter 10.1 (min/max), read at home. Next time we will go to Chapter 7 (Heaps).

Problem: Find the i-th smallest element (Rank-i).
- i=1 Minimum
- i=n Maximum
- i=n/2 Median

Possible solution:
- Sort
- Index into A(i).

We can do better!

\[ O(n \log n) \]

Lecture 5, Tuesday 4/17/01

Randomized selection

Divide and conquer approach:
- RS(A,p,r,i)
  - if p==r then return A(p)
  - q=RandomPartition(A,p,r)
  - k=q-p+1
  - if i<k then return RS(A,p,q-1,i)
  - i>k then return RS(A,q+1,r,i-k)
  - i==k then return A(q)

Correctness:
- Assume correct for size at most n=r-p+1
- after the partition, the arrays are smaller than n; can apply induction.
- Claim: need to search only one part
- Explain the 3 cases.

Lecture 5, Tuesday 4/17/01

Performance of Random Selection

Lucky case:
- What if 99/100 instead of 9/10 ??

Bad case:
- \[ T(n) \]
- Condition on partition outcome:
  - Substitute \( n \), choose \( c \) large enough for \( T(1) \):
  - \( T(i) \)
  - \( \sum_{i=1}^{n} \left( \frac{n}{k} \right) \)
  - \( \frac{n}{k} \)
  - \( \Theta \)
  - \( \leq \)
  - \( \frac{n}{k} \)
  - \( \leq \)
  - \( \frac{n}{k} \)
  - \( \Theta \)

Analysis continued

Let \( T(n) \) be the expected running time.
Condition on partition outcome:
- \( T(i) \)
- \( \sum_{i=1}^{n} \left( \frac{n}{k} \right) \)
- \( \frac{n}{k} \)
- \( \Theta \)
- \( \leq \)
- \( \frac{n}{k} \)
- \( \leq \)
- \( \frac{n}{k} \)
- \( \Theta \)

Deterministic Order Statistics

The randomized order statistics is very fast in practice (just like quick-sort, some additional tricks will help).

Theoretically interesting question: Is there a deterministic linear time order-statistics algorithm?

Deterministic selection algorithm (select i-th smallest):
- Divide \( n \) elements into groups of 5.
- Find median in each group (brute force)
- Use select recursively to find median among \( n/5 \) medians.
- Partition around this median.
- Recurse on the "appropriate" part, update \( i \) if necessary.

Deterministic order statistics - cont

Correctness - as before. All we changed was the pivot choice.

Time:
- At least \( 1/2 \) of the medians are \( \leq c \):
- each median brings 3 elements
- total
- \( n \geq \frac{3n}{2} \) for \( n \geq 6 \), we have
- \( n \geq \frac{3n}{2} \)
- at least \( n/4 \) elements are \( \leq c \).
- Similarly, at least \( n/4 \) elements are \( \geq c \).

Recursion:
- \( T(1) \)
- \( T(\frac{n}{2}) \)
- \( T(\frac{n}{2}) \)

In fact, we have \( T(1) = O(n) \)
Deterministic selection

- Homework:
  analyze with groups of 4 elements and groups of 6 elements.

- Observe that we can get deterministic variant of quicksort!
  - Can use as a black-box $O(n)$ partitioning into 2 equal parts.
  - We get recurrence $T(n) = 2T(n/2) + o(n)$, giving us $o(n \log n)$ total running time.
  - (do you think it will work well in practice?)