→ Handouts (4)

6.046
02/07/2012
L1.1

6.046: Design and Analysis of Algorithms

People

Prof. Bruce Tidor 32-212 tidor@mit.edu TH 8:30-9:30 am & by appt.

Prof Dana Moshkovitz 32-6606 dmoshkov@mit.edu H 1:30-2:30 pm & by appt.

Teaching Assistants

MT 4:00-5:00 (room TBA)

office hour updates on course website

→ Class Mechanics

Communications

Website ("Stellar")
- handouts
- lecture notes
- upload homework
- announcements
- office hours
- (calendar)

Be sure you have access to Stellar.

Make account on Piazza or link 6.046 to existing account.

Discussion threads ("Piazza")

email: 6046-tas@mit.edu

→ Class meetings

Lecture Tu Th 9:30-11:00 here → notes posted by are not a complete presentation.

Recitation F → required - new material and review lecture & hw material

- recitations meet this week
- registrar should be scheduling you for recitation. If not, attend what you would like.

→ Prerequisites

① 6.006 Introduction to Algorithms, AND

② [6.042/18.062J Mathematics for Computer Science OR
18.310 Principles of Applied Mathematics

→ with C or better grade. Talk to us if this isn't the case.
Of course we will assume knowledge of this material.

→ Textbook "Introduction to Algorithms" (3rd ed.) – additional resources

→ Problem Sets

- Nine problem sets
- Out on Wednesday and due the next Wednesday
- Distributed through Stellar and returned as pdf uploaded to Stellar
- Specific instructions in the handout — name collaborators
- Generally consist of ~2 problems to hand in as well as a few ungraded exercises that aren't handed in, but teach important material.
- late policy
  - no late homework, unless
  - can use up to 3 "grace days"
  - prior arrangement, for extenuating circumstances, and usually with Dean's excuse.
- "Exponential" Penalty for missing PS questions

→ Writing up homework — Clarity and precision of meaning is important.

- Being correct isn't really valuable if it isn't understable
- Simple elegance is a virtue and is worth more points
- "Give an algorithm" = Essay
  1. Describe in English, with pseudocode if helpful
  2. A worked example or diagram to illustrate
  3. Correctness proof
  4. Analysis of running time

Quizzes

Quiz 1 – In class – Tuesday, March 6

Quiz 2 – Take-home – Monday, April 9 (9am) → Fri, April 13 (noon)

→ Final Exam

"Grace Days" cannot be used

Grading Policy

Problem sets	20%
In-class quiz	20%
Take-home quiz	25%
Final Exam	30%
Recitation Participation	5%

Penalty for homework problems not attempted

#	penalty
0	None
1	1/100th letter grade
2	1/10th letter grade
3	1/5th
4	1/4th
5	1/3rd
6	1/2
7	1 letter grad
8	2 letter grades
9 or more	Fail

- problems, not Psets
- A problem will be considered skipped if no significant attempt was made to solve it.

Collaboration Policy

Many people find they learn more from the homework if work with peers. Others do better alone.

- Attempt yourself first
- What you hand in must be your own write up — without assistance and certainly not copied.
- You should be able to explain your solution to the staff
- Write collaborators on Homework
- No collaboration on quizzes or exam

What is the study of algorithms, and why is it important?

→ Correctness - (think security)

→ Efficiency — asymptotic, scalability

- Demonstration to ourselves and others
- Communication to others

* Basis for Design — language & building blocks

→ Mathematics is sufficiently precise to assure correctness & running time proofs

→ Relationship to 6.006

- try to avoid overlap, but 6.046 builds on 6.006
- 6.046 is generally more advanced
- 6.046 has no programming

→ Topics

- Divide-and-Conquer
- Dynamic Programming
- Greedy Algorithms
- Graph Algorithms
- Randomized Algorithms
- Data Structures
- Approximation Algorithms

Today – Review ideas you're familiar with, using an example that may be new

Order Statistics ("Median Finding")

Problem: Find the i^{th} smallest of n elements

ex: $i=1 \Rightarrow$ minimum

$i=n \Rightarrow$ maximum

$i=\lfloor (n+1)/2 \rfloor$ or $\lceil (n+1)/2 \rceil \Rightarrow$ median (convenient if n is odd)

→ Important for accessing databases and other problems

Intuitive Approach:

① Sort the set of n elements —merge sort→ $\Theta(n \lg n)$

② Jump to i^{th} element ⟶ $\Theta(1)$

$+$

$\Theta(n \lg n) + \Theta(1)$

$\Downarrow$

$\Theta(n \lg n)$ worst-case

Is there a better approach?

→ Don't try to find it, just think about whether there should be one!

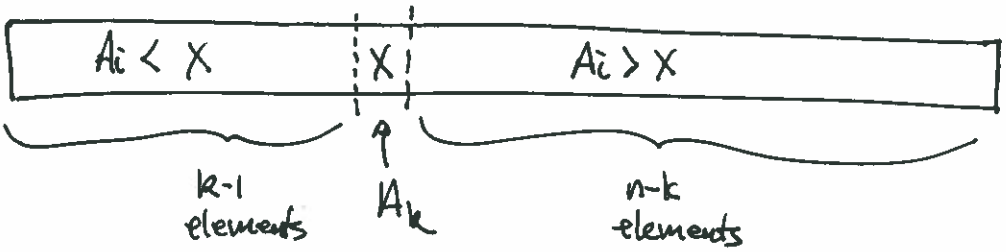
- We solved a harder problem than asked — because array is ordered, have i^{th}, $(i+1)^{th}$, etc., which we weren't asked for.
- So just because we reduced the problem to one for which we know the running time — haven't found running time for our problem necessarily.

Worst-case linear time deterministic algorithm (§9.3)

→ Helpful to assume elements are distinct (no duplicates)

Select (i)

1. Divide the n elements into groups of 5 (plus any remainder)
2. Find median of each group of 5 (by rote)
3. Use Select recursively to find median x of $\lceil n/5 \rceil$ medians
4. Partition elements around x. Let k = rank(x) → more later in course



5. If i=k, then return x.
   elseif i<k use Select recursively to find i^{th} smallest element on low side of partition
   else (i>k) use Select recursively to find $(i-k)^{th}$ smallest element on high side of partition.

Characteristics of Algorithm
- recursive
- non-obvious
- does "less work" than full sort — may be more efficient

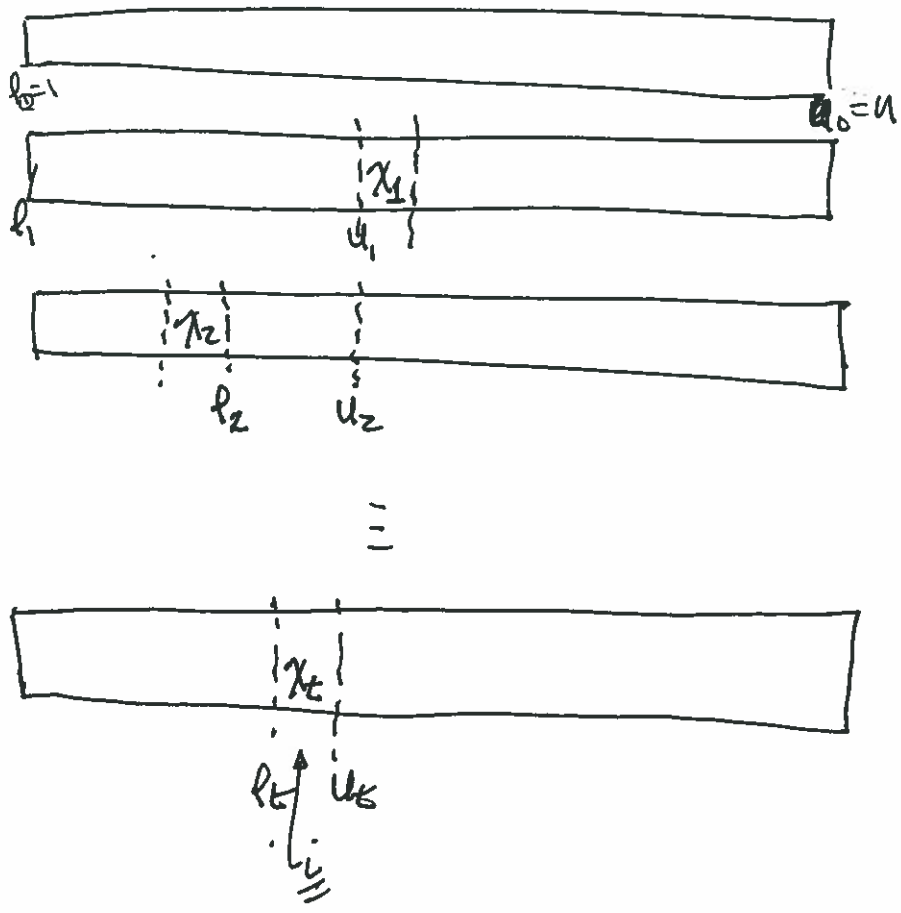
How should we prove CORRECTNESS?

→ Mix of knowing standard approaches and inventing new ones when the don't hold

Loops often use loop invariances to prove correctness. → what does this mean?

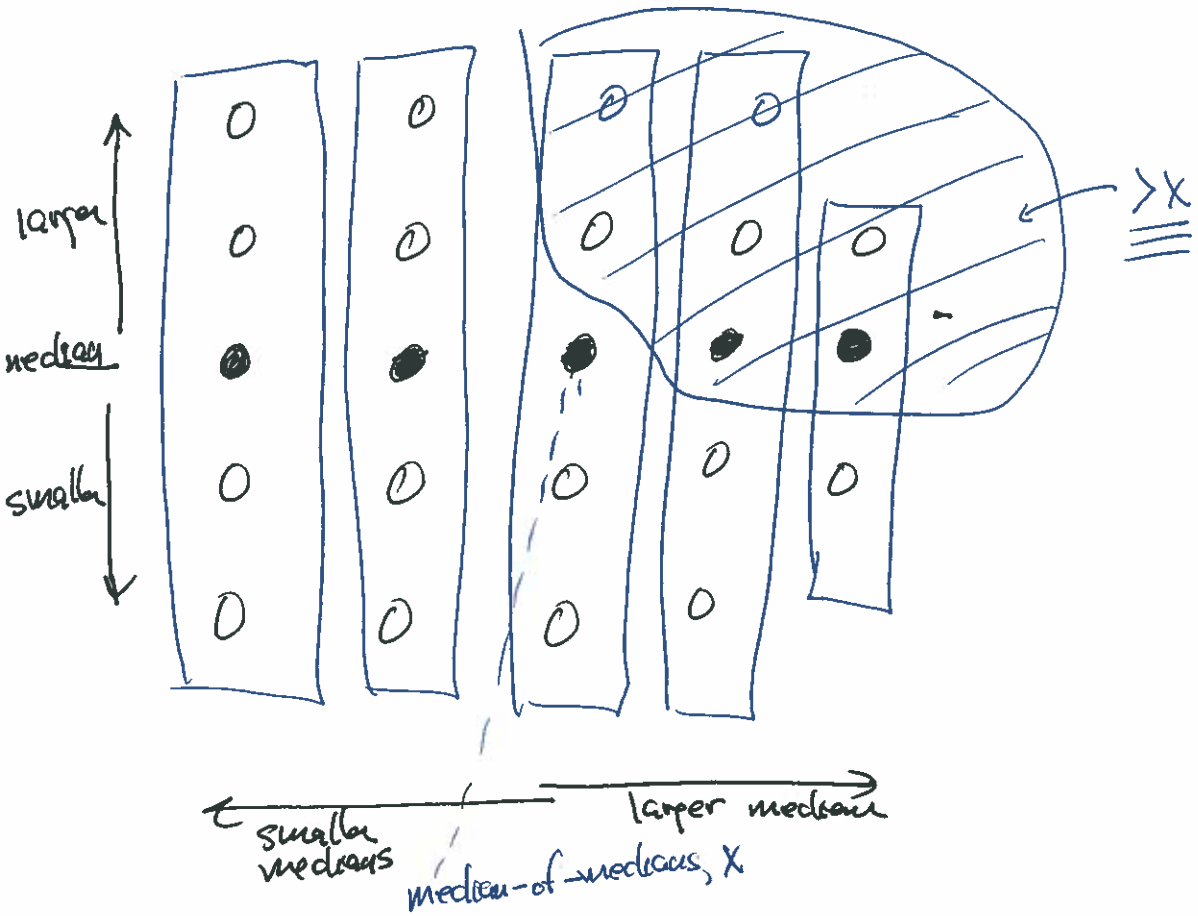
Set of conditions that are true when loop is initialized, maintained in each pass through the loop, true at termination → that helps prove correctness.

One possibility here is that at every pass through the loop, the "active subarray" consists of all elements of the original array bounded by the current min and max, and the current "index" is correct. → Think about this and other approaches.



- vague about locations
- messy up #s down
- messy partition step...

How to find and then prove running time?



- (At least) half of the medians from step 2 $\geq x$.
- (At least) half of the $\lceil n/5 \rceil$ groups contribute at least 3 elements $> x$, except for group with less than 5 elements and group containing x.

$$\#\text{ elements} \geq x \text{ is } 3\left(\left\lceil \frac{1}{2}\lceil n/5 \rceil \right\rceil - 2\right) \geq \frac{3n}{10} - 6$$

Symmetric argument...

$$\#\text{ elements} < x \text{ is also at least } \frac{3n}{10} - 6$$

Then, in worst case step 5 recursively calls Select with at most $\frac{7n}{10} + 6$ elements

Let's think about $T(n)$ — worst-case running time

Steps 1, 2, and 4 → what is their running time?

divide into groups of 5 — $O(n)$

find medians of $\lceil n/5 \rceil$ groups of (at most 5) — what if used insertsort? Each is $O(1)$ because 5 doesn't grow with n, so total is $O(n)$

partition — didn't show you (chpt 7), but $n-1$ comparisons, and so turns out can be done $O(N)$.

Step 3 takes $T(\lceil n/5 \rceil)$ — finding median of $\lceil n/5 \rceil$ medians

Step 5 takes at most $T\left(\frac{7n}{10} + 6\right)$

So our recurrence is

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

How do we go about solving recurrences?

1) Substitution Method — guess & prove by induction
2) Recursion-Tree Method — branching analysis
3) Master Method — pre-solutions

Substitution Method

- to deal with base case, often need to separate out small n solns
- here there is an additional reason (see below: $\lfloor n-70 \rfloor$ division)
- while the value 140 is arbitrary, it is convenient, and we need something at least ≥ 70
- why this is useful becomes clear below

$$T(n) \leq \begin{cases} \Theta(1) & \text{if } n < 140 \\ T(\lceil n/5 \rceil) + T(\frac{7n}{10}+6) + \Theta(n) & \text{if } n \geq 140 \end{cases}$$

Assume linear: $T(n) \leq cn$

→ Inductive case

$\Theta(n)$ term bounded above by an

$$T(n) \leq c(\lceil n/5 \rceil) + c(\tfrac{7n}{10}+6) + an \quad \leftarrow \text{substituting into inductive hypothesis}$$

$$\leq (\tfrac{cn}{5}+c) + (\tfrac{7cn}{10}+6c) + an$$

$$= \tfrac{9cn}{10} + 7c + an$$

$$= cn + \underbrace{\left(\tfrac{-cn}{10} + 7c + an\right)}_{\text{if } \leq 0} \Rightarrow c \geq \frac{10an}{n-70} \quad \text{for } n > 70$$

$$\leq cn$$

Because we chose $n \geq 140$, then $\frac{n}{n-70} \leq 2$ (a convenient integer), and letting $c \geq 20a$ will satisfy our inequality.

→ Base case

$$T(140) \leq T(\lceil 140/5 \rceil) + T(\tfrac{7 \cdot 140}{10}+6) + \Theta(n)$$

$$\leq \Theta(1) + \Theta(1) + an$$

$$\leq b + b + an$$

$$= 2b + an = 2b + 140a \leq cn = 140c$$

$c \geq a + \frac{b}{70}$ satisfies the base alone

Choosing $c \geq 20a + b$ will satisfy base and inductive cases. □