Today: Take home
- format & rules
- academic honesty
- how to do well

Format: 5 problems, equally weighted
(no blank answer = 30% rule)
- solutions submitted separately via Stellar
  (like psets - typeset or scan)

Deadline: 4:59 pm on Monday, April 22
- exceptions only via Dean's note
Rules: just like an in-class quiz
(except you can leave the room)
- cannot communicate with anyone except the course staff (6046-staff@csail.mit.edu) about any aspect of the exam until April 30 (even if you've turned it in)
(or until we say) (they may not have)
  - don't say "how's the quiz going" or "tough Problem n" ~ easily leads to worse
- limited open book:
  - your course notes (including psets)
    (also your 6.006 & 6.042 notes)
  - CLRS textbook
  - our course website
  - basic references: dictionaries, calculators
    (& Python & Mathematica ~ but don't waste time)

NOT ANYTHING ELSE:
- other offerings of algorithms courses
- other algorithms books
- rest of web

If you may have made a mistake, contact the course staff immediately.
(better you than us or someone else)
Psychology of cheating:

① Imposter Syndrome
- how did I get into MIT?
- everyone's smarter than I am
- need to make them think I belong...

RELAX: you all belong
- you don't have to well in Quiz 2 for decent grade, or life in general
- personally, one of my hardest classes (but also why I like it)

② Bunker Mentality
- war (psets, exams,...) reigns down
- bunkmates (fellow students) become your world ~ close bonds
- distorted sense of morality

RELAX: quiz is about learning & proving YOURSELF ~ not survival
Why not to cheat:

**Educationally:**
- great test of algorithmic creativity
- challenge & learn about yourself
- everyone wants this exam to be fair

**Morally:**
- we trust you to do the right thing
- keep MIT integrity high

**Practically:**
- we are good at finding cheating
- we will follow through
- you will fail & may get expelled

**Just say no to cheating**
How to do well:

**Conceptually:**
- every question asks for an algorithm
- goal: fastest (correct) algorithm
- worst case \( \Rightarrow \) w.h.p. \( \Rightarrow \) expected amortized

**Writing:**
- goal: clear & concise
- should start with an **executive summary:**
  - problem you are solving
  (and any assumptions you made)
- techniques you’re using to solve it
- running time bound
- English description of algorithm
- be explicit about steps
  - e.g. don’t just say “sort” ~ say how
- use pseudocode only if necessary
  - to clarify your solution
- prove correctness (succinctly & convincing)
- analyze running time
  - treat all unknowns as parameters
- cite algorithms/theorems from class/CLRS
- don’t include irrelevant facts
Bugs:
- if we find one, we will email you 
  \(\Rightarrow\) check your email daily
- if you find one, email 6046-staff@csail.mit.edu

Planning: \(\sim 12\) hours in 4.5 days
- get plenty of sleep
- work on all problems today
- creativity takes time
- don't get sucked into one problem
- budget time for writing & debugging

Strategy:
- start with a (weak) upper bound  
  \(-\ e.g.\ exponential\)
- successively improve, writing up current best solution
- look for analogies with problems you've/we've solved in class
- try all the design techniques we've covered
- try solving special cases
- work out small examples

Relax, be positive, be persistent.
Don't cheat.