Notes on the Midterm

- **Value:** 25% of your grade

- **Format:**
  - online, closed book
  - you will be allowed to use one A4 page of notes (front and back). If you type it, but please avoid too small fonts.
  - you can write, type (with your favorite editors) or mix them for your submission.
  - scan/take a picture and convert your solution to pdf for submission. Main thing is being clearly readable.

- **When:**
  - 2:00pm (Thursday, Feb 17)
  - Questions available at 1:45pm on Canvas (as an assignment).
  - 1:30 hour for the exam, and 20 mins to scan and submit
  - submit before 3:50 pm

- **Helpful Resources:**
  - Our slides and your own lecture notes
  - Questions in our quizzes and assignments are good examples
Possible topics in Midterm

• Essentially everything discussed before our class in Feb 10 (i.e., not covering hash tables), but the following topics are focused:
  • Asymptotic notations, their definitions, and their relations;
  • Relations between functions using asymptotic notations;
  • Runtime of loops. Given a loop or nested loop, determine its asymptotic runtime as a function of some variable \( n \) using asymptotic notations.
  • Loop invariants: You will be given a loop invariant, \( \alpha \), and you will use it to prove the correctness of a loop. (Coming up with loop invariants is an important skill, but not one that I will evaluate you on in a timed midterm.);
  • Linear data structures (i.e., arrays, linked lists, stacks, queues), possible implementations for their operations, and corresponding running time;
  • Priority queues (abstract data type) and implementation as sorted or unsorted list.
  • Binary heaps: their properties (e.g., heights, index rules for child/parent), representations (e.g., array vs tree), and implementations of their operations (e.g., insert, extract_max, increase_key, heapsort) and running time;
  • Designing efficient algorithms for some problems using the data structures we discussed so far.
Not covered in Midterm

• Formal proofs for asymptotic statements
• Formal proofs of amortized analysis
• Pre-order, in-order, post-order traversal of trees
• Ordered tree representation
• Hash tables