1 Important Midterm #2 Topics

Midterm #2 topics (this list is not exhaustive):

- Unbounded arrays
- Amortized analysis (*We’ll go over an example of this today*)
- Hash tables (Hash functions, separate chaining, * probing)
- Interface vs Implementation
- Priority queues
- Heaps
- Data structure invariants and restoring invariants
- Binary search trees
- AVL Trees
- `void*` polymorphism
- Memory management
- C
- Material from midterm #1 (exam is cumulative)

Remember the material from midterm #1? Contracts, loop invariants, bitwise operations, arrays, linear search, binary search, proofs, Big-O notation, unit testing, structs, pointers, linked lists, memory layout, recursion.

**Question:** What topics are you most uncomfortable with? Anything in particular you’d like to see in recitation today?
2 Amortized Analysis: Binary Counter

Consider an $n$-bit binary number. Assume that flipping a bit (changing it from a 1 to 0, or 0 to 1) is a constant-time operation.

Question: What is the worst-case time complexity of incrementing the number, in terms of $n$?

Question: What is the amortized time complexity of incrementing the number, in terms of $n$? (What are the two ways we can conceptualize this analysis?)

Strategy: Be sure to define what each of the variable in your big O notation stand for ahead of time!

Thanks to Josh Zimmerman for the formulation of this problem in recitation 12.

Question: Do you recall the amortized analysis we did for unbounded arrays?

3 Your Pick

What would you like to see more of?