Honors CS 2 Exam #1 Review

Date: 10/8/2026 (Thursday)
Place: BHC-129
Time: 9:00 – 10:30 am

Outline of material covered so far:

I. Java Tools: Built in Sorting, Data Structures
- **a. How to use Arrays.sort, Collections.sort**
- **b. ArrayDeque**
- **c. PriorityQueue**
- **d. HashSet**
- **e. TreeSet**
- **f. HashMap**
- **g. TreeMap**

II. Backtracking
- **a. Just Recursion**
- **b. Stop traversing a path when it's doomed to fail**
- **c. Basically, "try all reasonable possibilities"**
- **d. K-divisibility**
- **e. Eight Queens**
- **f. Magic Squares**
- **g. Tentaizu**
- **h. Paintings**

III. Disjoint Sets (Not on test)
- **a. Array representation**
- **b. FindSet Operation**
- **c. Union Operation**
- **d. Path Compression**
- **e. Use in tracking components, component sizes**
- **f. Severing connections can be building them in reverse.**

IV. 2-4 Trees (Not on test)

- **a. 2-4 Tree Node Property, Leaf Node property**
- **b. Insert**
- **c. Delete**
- **d. Tracing only**

V. Red-Black Trees (Not on test)

- **a. R-B node properties**
- **b. Insert**
- **c. Delete**
- **d. Equivalence with 2-4 Trees**
- **e. Tracing only**

VI. Binary Index Trees

- **a. Internal Storage for table of responsibility**
- **b. Use of lowest one bit**
- **c. O(lg n) point update, range sum query**
- **d. Use in counting array inversions**
- **e. Application in program 3**

VII. Summations/Algorithm Analysis

- **a. CS1 Summation Rules**
- **b. Arithmetic Sums**
- **c. Geometric Sums (finite, infinite)**
- **d. Multiplying equations and subtracting idea**
- **e. Discrete Random Variables and Avg. Case Runtime**

VIII. Sorting/Searching

- **a. Quick Sort Analysis**
- **b. Quick Select Analysis**
- **c. Proof of Comparison Sort Lower Bound (details)**
- **d. Bucket Sort**
- **e. Counting Sort**
- **f. Radix Sort**

IX. Greedy Algorithms

- **a. Coin Changing**
- **b. Room Scheduling (Single, Multiple)**
- **c. Containers Problem**
- **d. Planting Trees, Sushi Problem (do longest stuff first)**
- **e. Use of Exchange Argument to Prove Correctness**
- **f. Huffman Coding**

X. Hybrid Algorithms (Not on test)

- **a. Welcome Party**
- **b. Walls**

Note: I normally don't tell students what's not on the test, I am unsure why I decided to this time. I am certainly likely to try to test this stuff in the future one way or another.

Format of exam:

You will have some short answer questions, some tracing questions, some coding (Java API, backtracking) and some creative problem solving based on what we've done. A large document with quite a bit of Java API documentation will be given. Here is a link to it:

https://www.cs.ucf.edu/~dmarino/ucf/cop3330/JavaAPI-Reference-ForQuizExam.pdf

How to study:

1) Look over the notes, paying attention to all the code shown in class. Make sure you understand how it all works.

2) Look over your programming assignments, making sure you remember how you solved certain problems.

3) Look over past exams on my archive.

4) Code up short examples related to backtracking, or trace through data structures, checking results with posted programs or websites that provide traces.