

CS 2 Exam #1 Review

Date: 2/10/22 (Thursday)

Place: CB2-207

Time: 3:00 – 4:30 pm

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. Sudoku**
- e. Magic Squares**
- f. Eight Queens**
- g. Tentaizu**
- h. Hexagram**

III. Disjoint Sets

- 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

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

V. Red-Black Trees

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

VI. Skip Lists

- a. Use of probability**
- b. Link structure**
- c. Insert**
- d. Delete**

VII. Programming Assignment Details (Posted Solutions)

- a. Politics**
 - i. Use of Hash Map with custom sorting**
 - ii. Array of Lists, arrays in order, lists in order.**
 - iii. Using long to store ordered pair of ints.**
- b. Tentaizu**
 - i. Recursive function takes in slot, number of bombs**
 - ii. Try two options (bomb, no bomb)**
 - iii. Cut out when a fixed square is wrong.**
 - iv. Idea of auxiliary storage (current adj bombs)**

Format of exam:

You will have some short answer questions, some tracing questions, some coding (Java API, backtracking) and perhaps some problem solving based on what we've done. Any necessary Java API documentation will be given on the exam (do not write the method names on your sheet of notes).

You may bring one 8.5"x11" piece of paper with notes as an aid to use during the test.

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.**