CS3230 Tutorial 6

Q1. Store the n numbers into a hashmap. O(n)

Then for all numbers search for V – A[i] in the hashmap.

Q2. Sorting – check if the upper bound is > then the next lower bound.

Question is asking, at one point, what is the maximum number of overlapping intervals, instead of what is the total number of overlapping intervals

* Sort first
* Bracket opening and closing, everytime open bracket + 1, closing bracket -1, find max value along anypoint

Q3. Start by creating a subtree of height 5

Q4. Assume that you have a max heap

* (1) Convert the max heap to min heap. O(n)
* (2) Swap the root element with last element, then sift down O(lg n)
* (3) If dunno position O(n), but O(lg n) if know position

Q5.

L1 + SUM(Sij) = L2 – series of operation – XOR or +1(commutative, order does no matter)

Covert matrix to row-major or col-major order

Graph transversal – light as a state

Q6.

EXPHORNER

// Array b[0..l] gives the

// ternary representation of n

p←a

for i from l − 1 downto 0 do

p←p×pxp

if b[i] = 1 then p ← p × a fi

if b[i] = 2 then p <- p x a x a fi

od

O(lg n = k)

Q7. Uninformed search. DFS – Maintain a goal state. Then DFS from initial state.