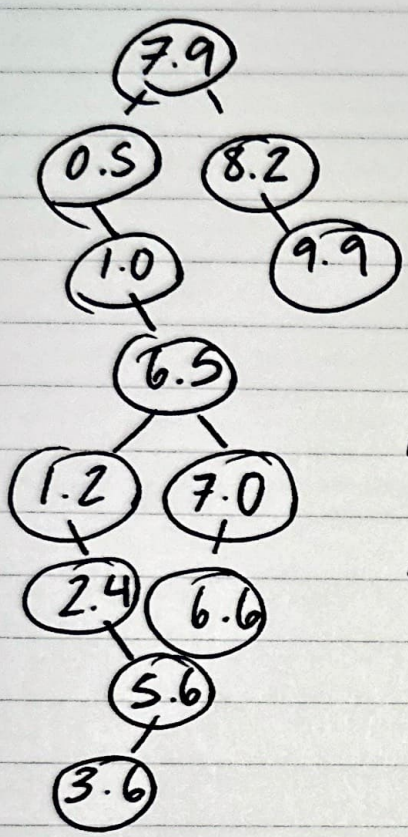
Trees + Heaps

Amalia Karaman

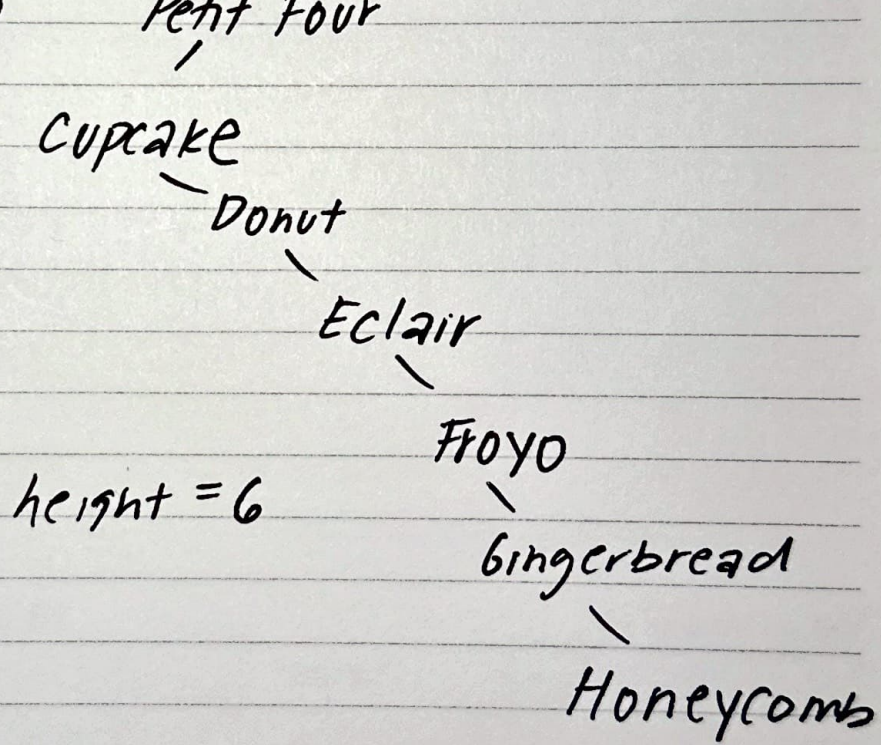
1)

2)



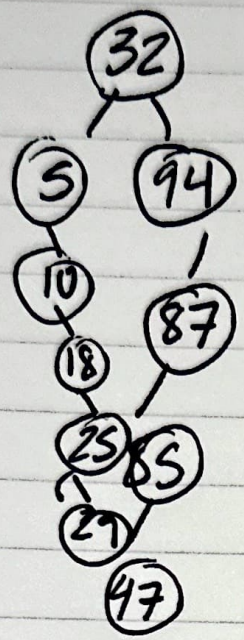
w/ height 7

b)



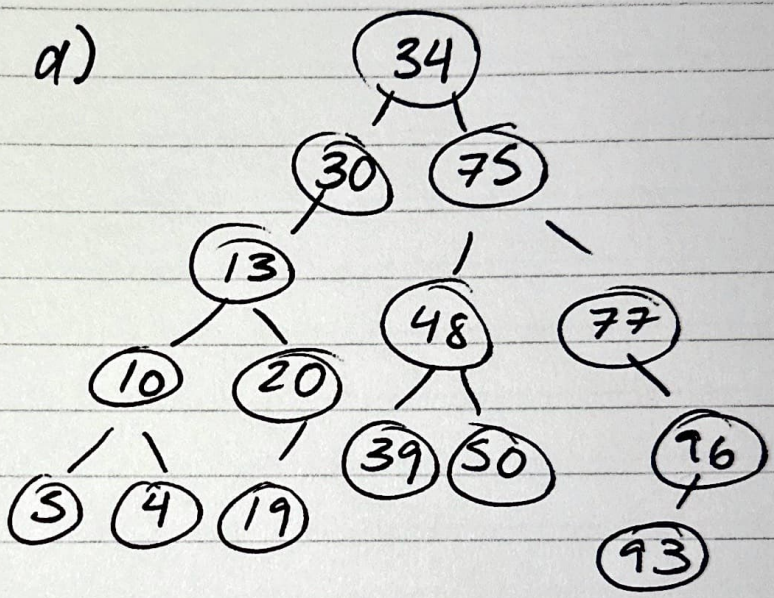
height = 6

c)



height=6

a)

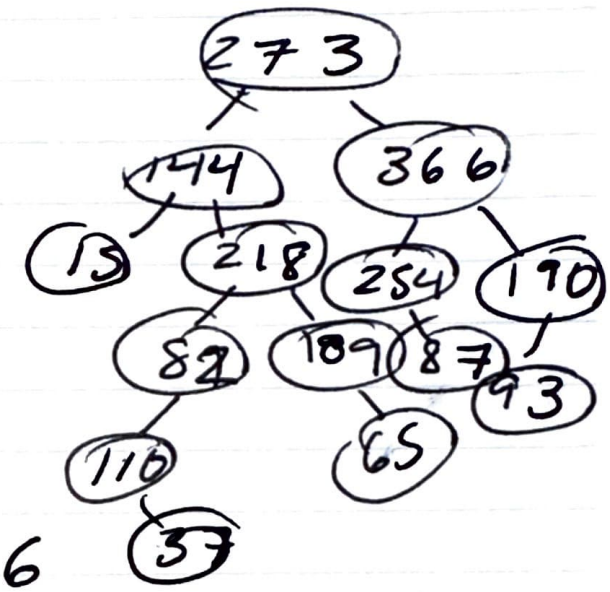


height = 6

2) Preorder:

node.data += left.data + (2 x right.data)

68 + 21 + (2 x 92) = 273
21 + 15 + (2 x 54) = 144
15 + 0 + (2 x 0) = 15
54 + 46 + (2 x 59) = 218
46 + 36 + (2 x 0) = 82
36 + 0 + (2 x 37) = 110
37 + 0 + (2 x 0) = 37
59 + 0 + (2 x 65) = 189
65 + 0 (2 x 0 = 65
92 + 80 + (2 x 97) = 366
80 + 0 + 2(87) = 254
87 + 0 + (2 x 0) = 87
97 + 93 + (2 x 0) = 190
93 + 0 + (2 x 0) = 93

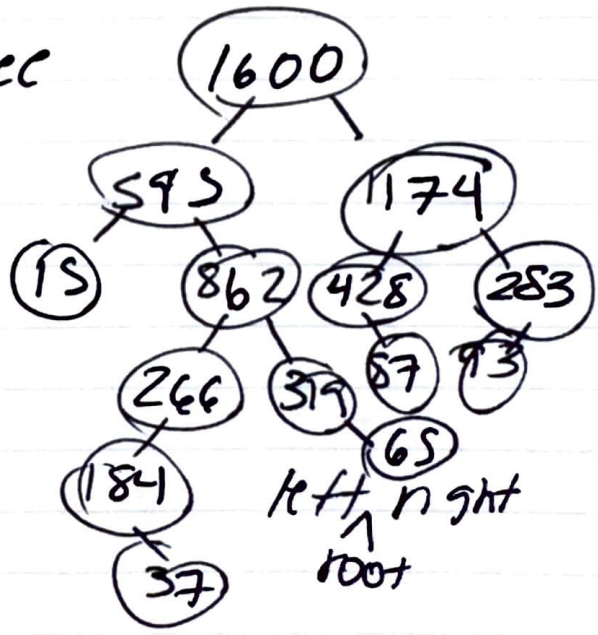


root, left, right

Inorder

15 + 0 + (2 x 0) = 15
144 + 15 + (2 x 218) = 595
110 + 0 + (2 x 37) = 184
37 + 0 + (2 x 0) = 37
82 + 184 + (2 x 0) = 266
218 + 266 + (2 x 189) = 862
189 + 0 + (2 x 65) = 319
65 + 0 + (2 x 0) = 65
273 + 593 + (2 x 366) = 1600
254 + 0 + (2 x 87) = 428
87 + 0 + (2 x 0) = 87
366 + 428 + (2 x 190) = 1174
93 + 0 + (2 x 0) = 93
190 + 93 + (2 x 0) = 283

tree



the

2b) No b/c since ~~[illegible]~~ values are no longer left root right b/c of the difference in arithmetic.

2c) No they aren't AVL's b/c AVL's require height balacing and the result wasn't rebalanced w/ their branches.

5) initializeCandidates(List<String> candidates)
Time: O(n) b/c loop runs once for each candidate → adds to map + heap
Space: O(n) b/c stores n candidates in map+heap
n = number of candidates

getTotalVotes(int p)
Time: O(1) ret an integer variable w/ no loops
Space: O(1)
stores 1 int w/ no growth

castVote
Time: O(log n)
map update is O(1) but inserting into heap is O(log n) b/c binary heap
Space: O(1)
modifies one value, adds one node, constant space usage

castRandomVote()
Time: O(n)
Space: O(n)

registerElection(String candidate)
Time: O(n)
Space: O(n)

getTopKCandidates(int K)
Time: O(n log n)
full sort of n entries → comparison-based sorting = O(n log n)
Space: O(n) stores sorted list of entries

auditElection()
Time: O(n log n)
Space: O(n)

Constructor & CandidateVotes
Time: O(1)
initializes empty data structures
Space: O(1)
no growth on creation :)