Q1: To proof $\Theta(\log n)$, we should proof $C_1 \log n \le f(x) \le C_2 \log n$. Since (2,4) tree can have 2–4 children node, so, $h_4 \le f(x) \le h_2$. for h_4: $1+4+\dots+4^{h-1} = \sum_{i=0}^{h-1} 4^i = n \Rightarrow \frac{4^h-1}{3} = n \Rightarrow h = \log_4(3n+1)$.
In the same way, $h_2 \Rightarrow h = \log_2(n+1)$. thus, $\log_4(3n+1) \le f(x) \le \log_2(n+1)$, so is $\Theta(\log n)$

INT202 Complexity of Algorithms

Tutorial 3

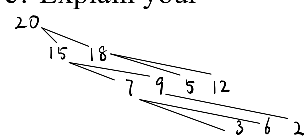
Q2:
a, k^h
b, $\sum_{i=0}^{h-1} k^i = \frac{k^h-1}{k-1}$

Q1. Show the height of a (2, 4) tree storing n keys is Θ(log n).

Q2. Let T be a k-ary tree, i.e. each internal node of T has at most k children. Suppose that the height of T is h (the depth of the root is 0).

a. What is the maximum number of external nodes that T can have? Explain your answer.
b. Compute the maximum total number of nodes that T can have.

Q3: heap was like:

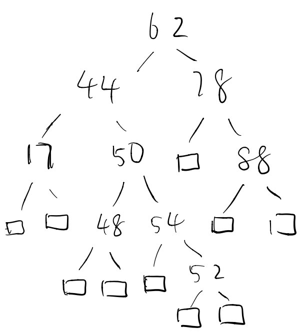


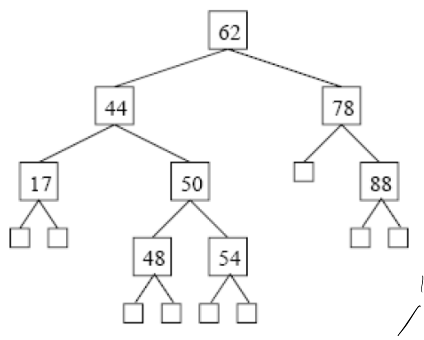
so, array is [20 15 18 7 9 5 12 3 6 2]

Q3. Show that array [20 15 18 7 9 5 12 3 6 2] forms a max-heap.

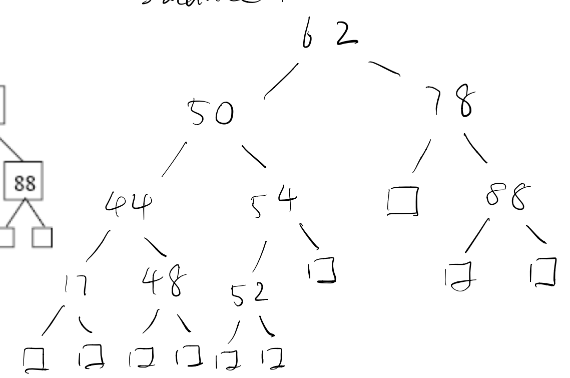
Q4. Draw the AVL tree resulting from the insertion of an item with a key 52 into the given binary tree. (Only the final tree is needed.)

Q4: after insertion:





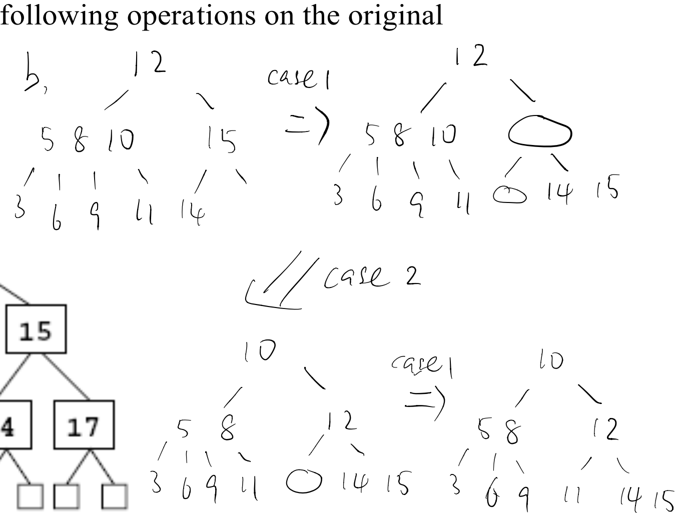
balance:



Q5. Let T be a (2, 4) tree shown below, which stores items with integer keys. Draw all the steps and the resulting trees obtained by performing the following operations on the original (given) T.

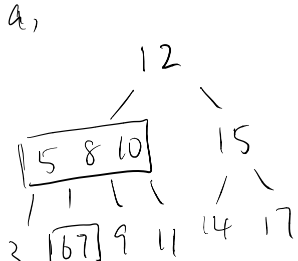
a. Insert an item with key 7 into T.
b. Remove an item with the key 17 from T.

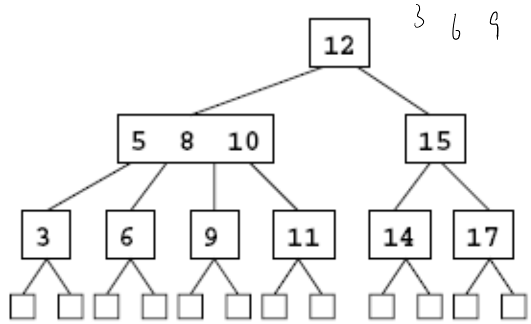
b,



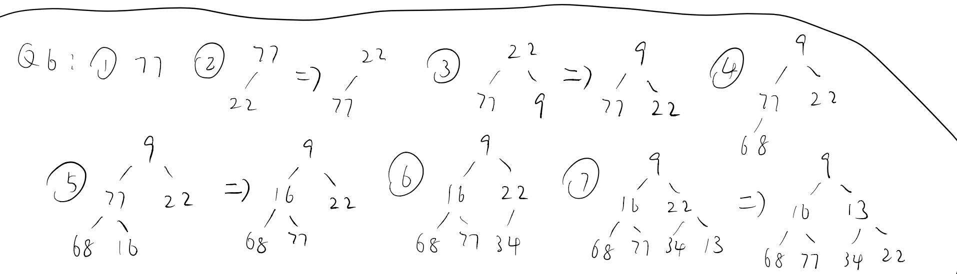
Q5:

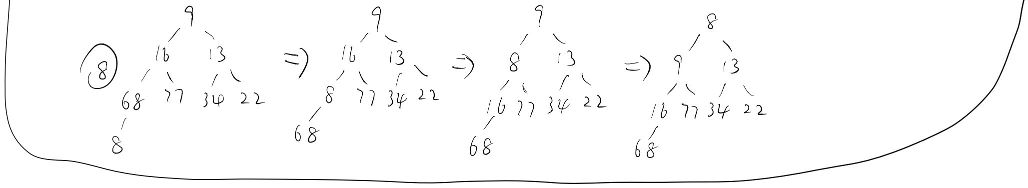
a,





Q6:





Q6. Draw the binary min-heap that results from inserting: 77, 22, 9, 68, 16, 34, 13, 8 in that order into an initially empty binary min heap. You are required to draw the intermediate heaps after each insertion and the final heap.

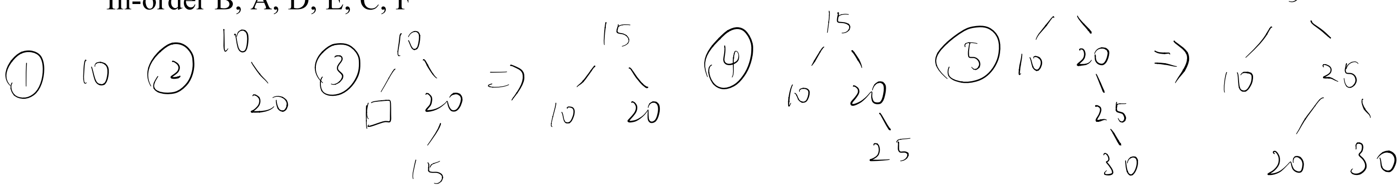
Q7. Insert the following sequence of elements into an AVL tree, starting with an empty tree: 10, 20, 15, 25, 30, 16, 18, 19. Draw the AVL tree after each insertion.

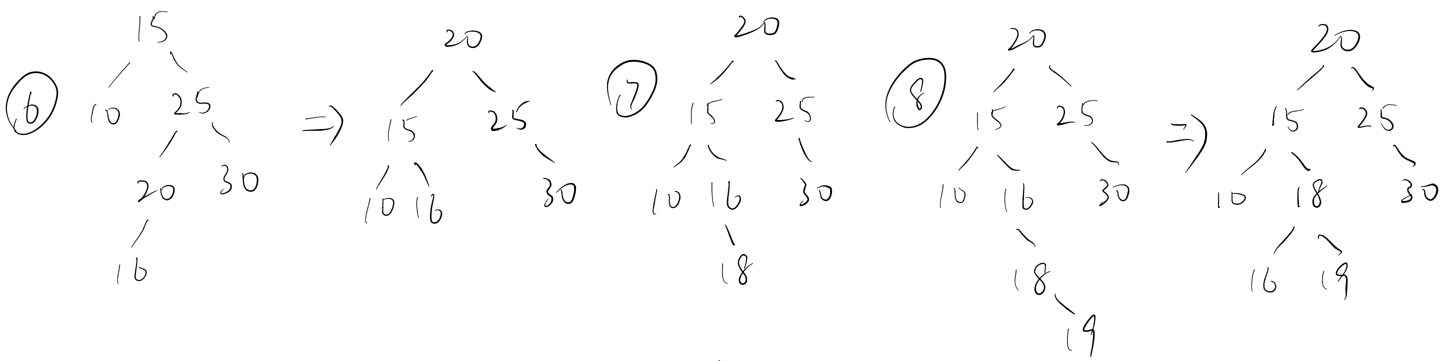
Q8. Given the following pre-order and in-order binary tree traversals, draw the binary tree.

Pre-order A, B, C, D, E, F

In-order B, A, D, E, C, F

Q7:





Q8: A is root, F is last