This problem set is individual and worth a total of 310 points. You do **not** need to submit anything for this problem set, the material is solely meant to help you better understand the course material and prepare for the exam.

Your Name: Eben Aceto

1. [5 points] Which of the three following options is the **worst** choice for a pivot for your partition function?

   (a) Always choose a random element to be your pivot.

   (b) Always choose element at index 0 to be your pivot.

   (c) Always choose the middle element to be your pivot.

2. [5 points] What makes this a poor choice of partition?

   It is too easy for it to yeild worst case performance given common inputs (reverse-ordered)

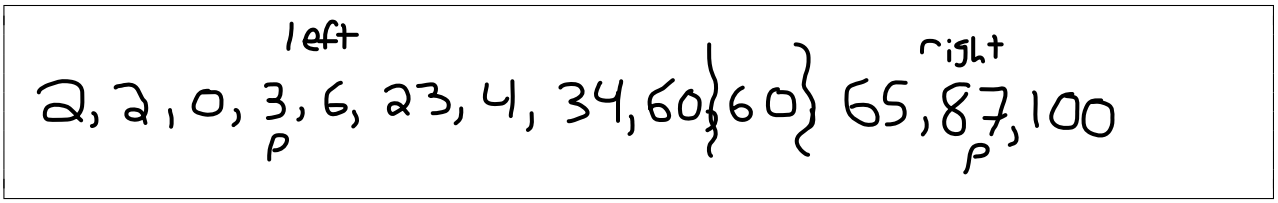
3. [15 points] Choosing the first element to be your partition and using the code in the course slides: Show the state of the following array after the partition function returns. Indicate the pivot points.

   ```
   [60, 100, 4, 2, 6, 23, 65, 34, 87, 3, 2, 0, 60]
   ```

   (a) After the first call to partition:

   3, 60, 4, 2, 6, 23, 0, 34, 2, 60, 87, 65, 100
   p

(b) After the second calls to partition:



4. [5 points] If we were to randomly choose an element from the above array to be the pivot point, which element would provide the most optimal(even) split of the array after the first partition?

23

---

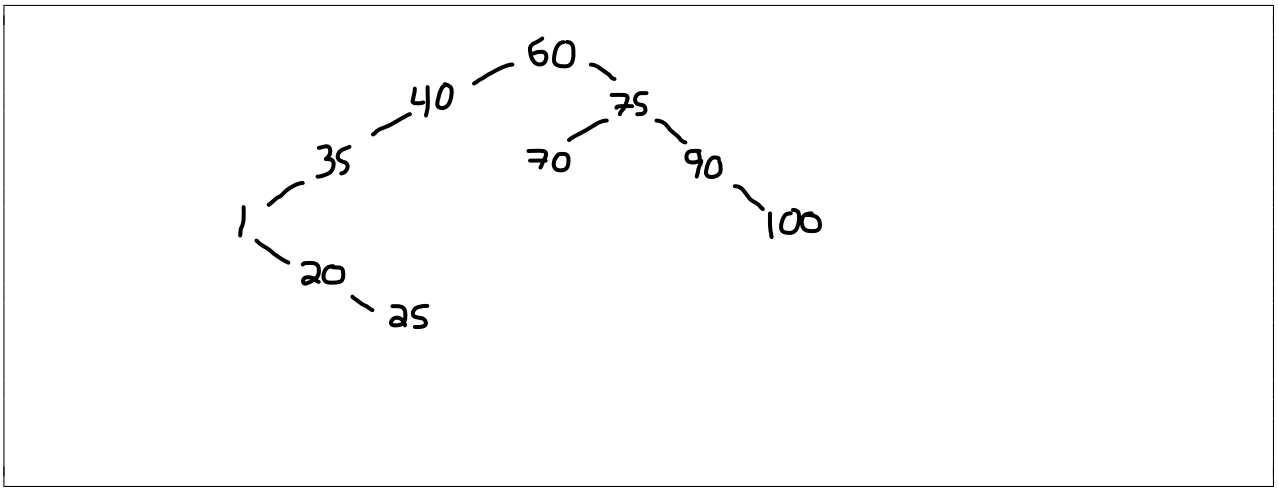
5. [5 points] What is the average big Θ runtime of quick sort?

$\Theta(n \log n)$

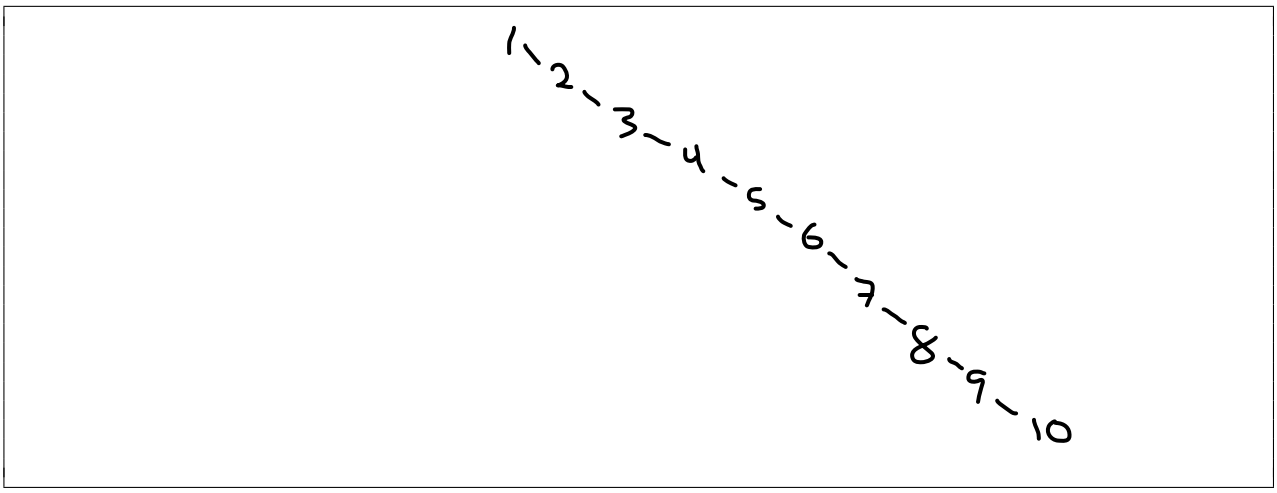
---

6. [15 points] Draw a binary search tree after the insertion of the following elements in order:

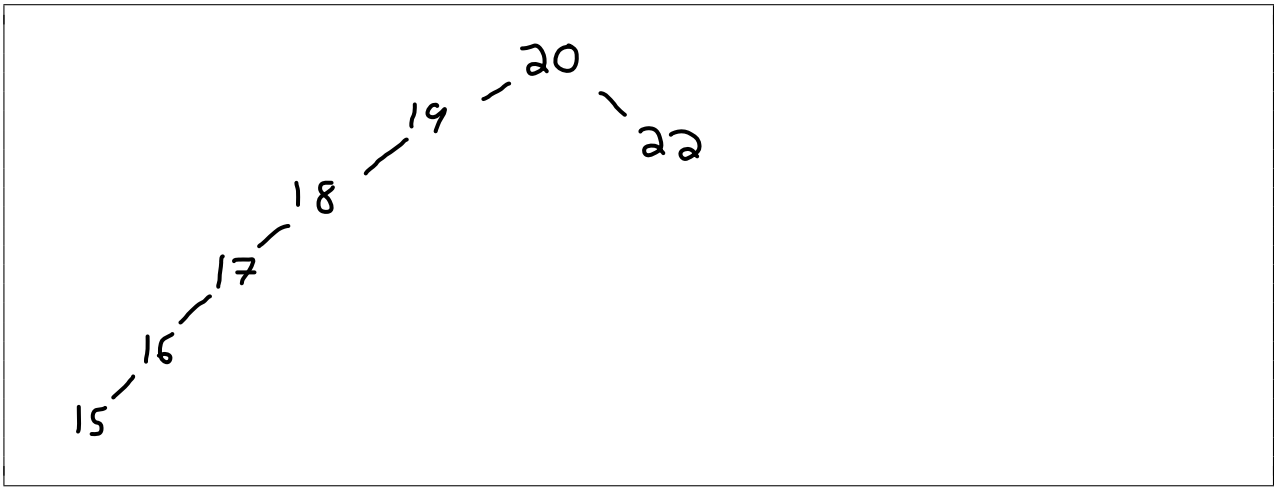
(a) [5 points] [60, 40, 35, 75, 90, 1, 20, 100, 25, 70]



(b) [5 points] [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]



(c) [5 points] [20, 19, 18, 17, 16, 15, 22]



7. [15 points] What do the following terms mean when referring to properties of k-ary trees?

(a) [5 points] Full (proper):

Every node has exactly 0 or K children

(b) [5 points] Complete:

Every level is entirely filled except possibly the deepest, where all nodes are as far left as possible

(c) [5 points] Perfect:

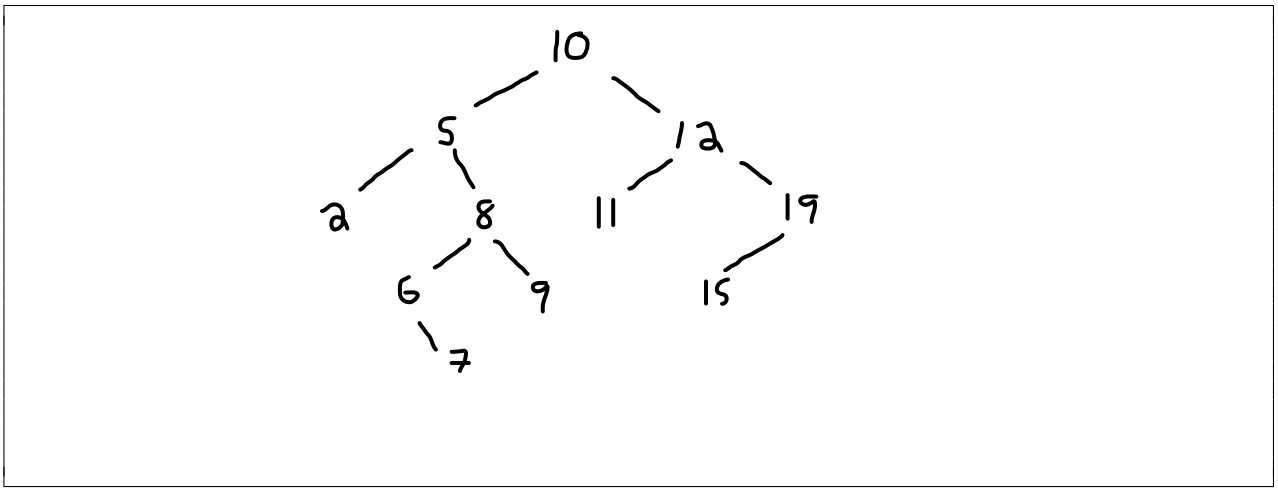
Tree is full and all leaves are the same depth

8. [5 points] If a binary tree is complete, does that necessarily mean it is also full? Justify your answer.

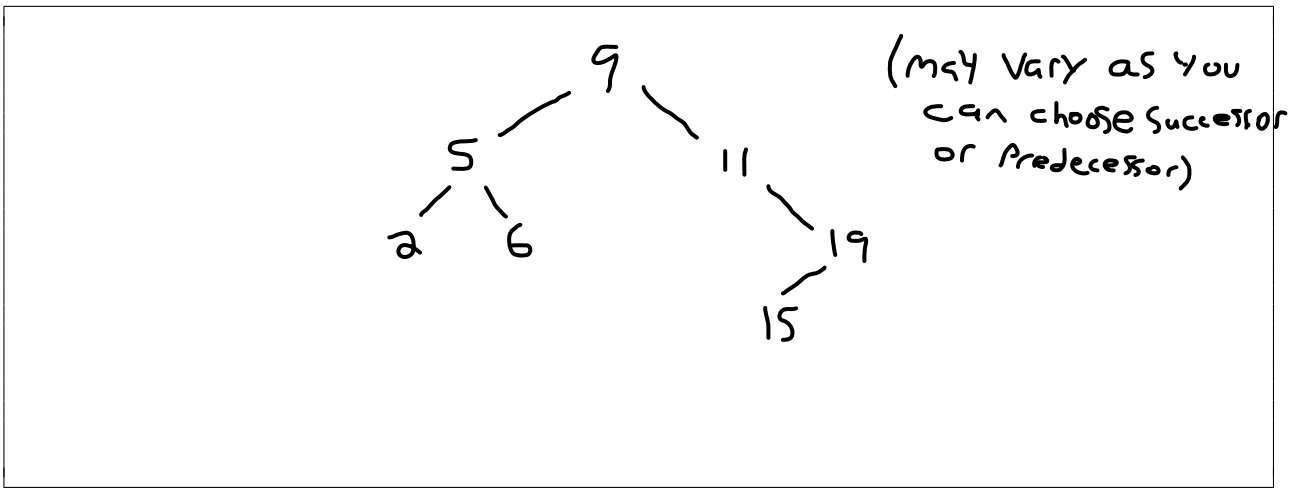
no, on the deepest level of a complete tree, there could be a node which does not have 0 or k children (see 13d)

9. [10 points] Draw an initially empty BST after performing the following operations:

(a) [5 points] Insert the following elements in order: 10, 5, 12, 8, 19, 6, 2, 11, 15, 9, 7



(b) [5 points] Remove the following elements from the tree created by part a: 7, 12, 8, 10



10. [15 points] Briefly explain the the following tree traversal methods.

    (a) [5 points] in-order traversal:

    Visit left, root, right

    (b) [5 points] pre-order traversal:

    root, left, right

    (c) [5 points] post-order traversal:

    left, right, root

11. [20 points] Let T be a full k-ary tree, where k = 2 (a.k.a. binary tree), with n nodes. Let h denote the height of T.

    (a) [5 points] What is the minimum number of leaves for T?

    $h+1$

(b) [5 points] What is the maximum number of leaves for T?

2^h

(c) [5 points] What is the minimum number of internal nodes for T?

h

(d) [5 points] What is the maximum number of internal nodes for T?

$2^h - 1$

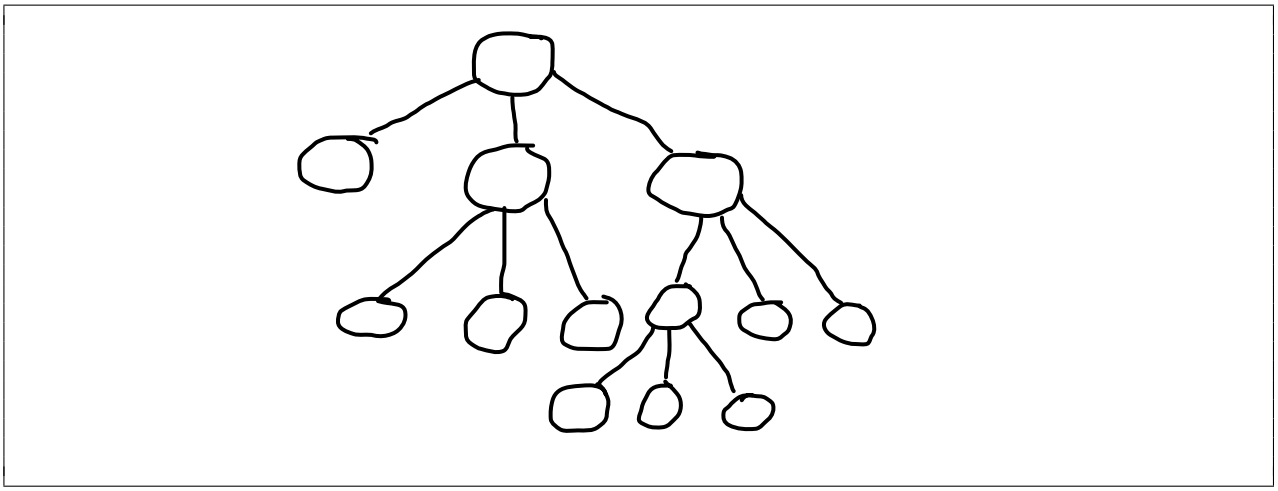
12. [10 points] Show that the maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.

A Perfect binary tree of height h has $2^{h+1}-1$ nodes,

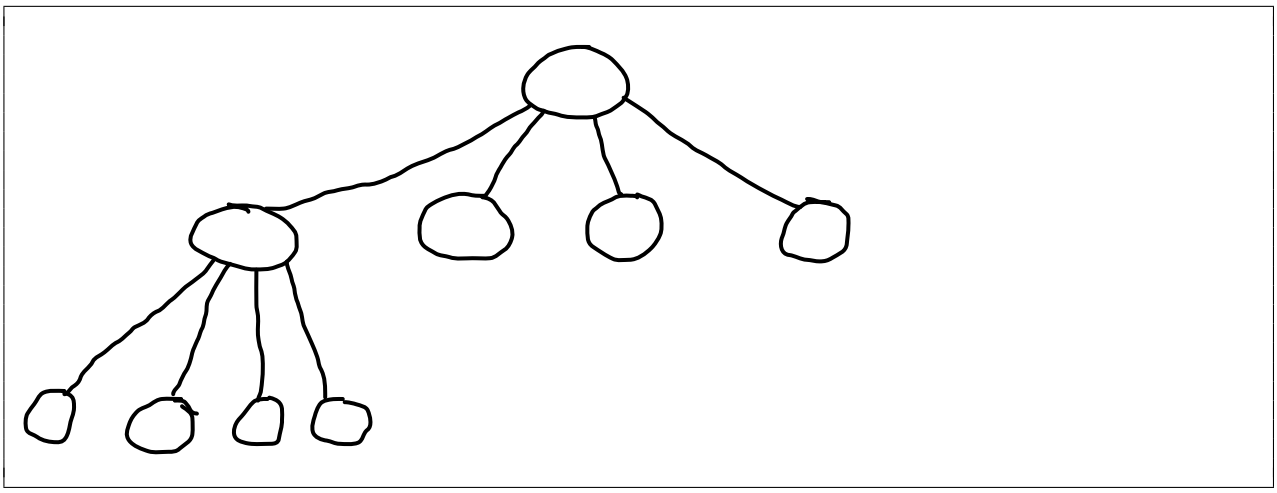
can't do better

13. [20 points] Draw an example of k-ary trees matching the following descriptions

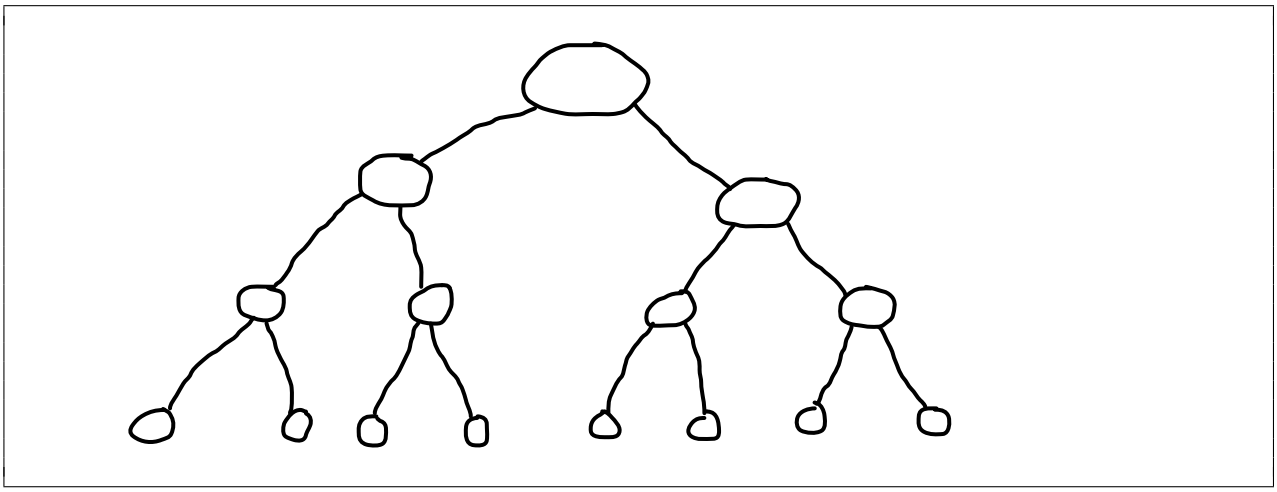
(a) [5 points] Full(proper), k = 3, height = 3



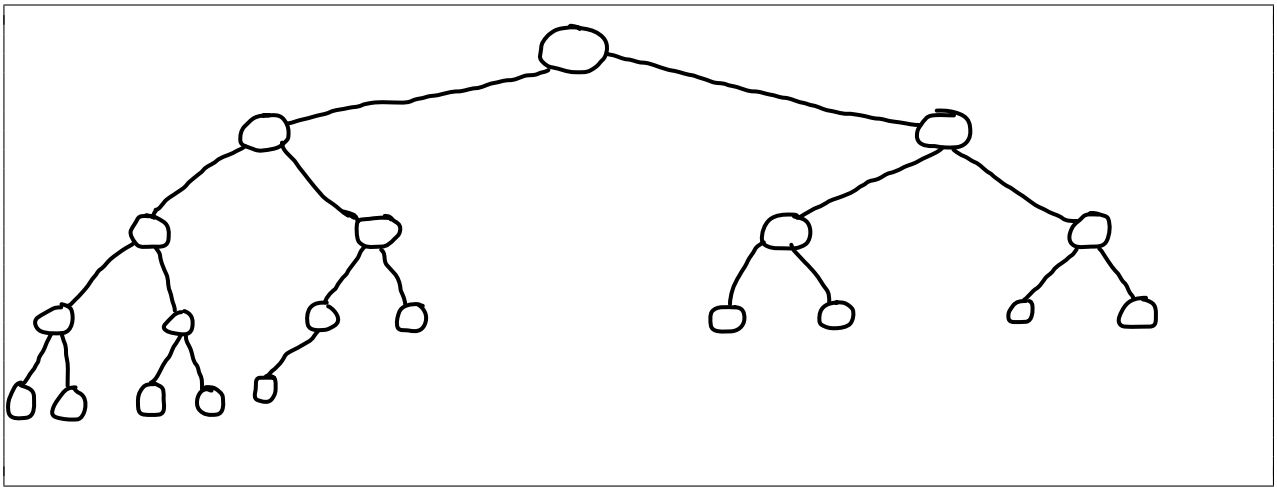
(b) [5 points] Complete, k = 4, height = 2



(c) [5 points] Perfect, k = 2, height = 3

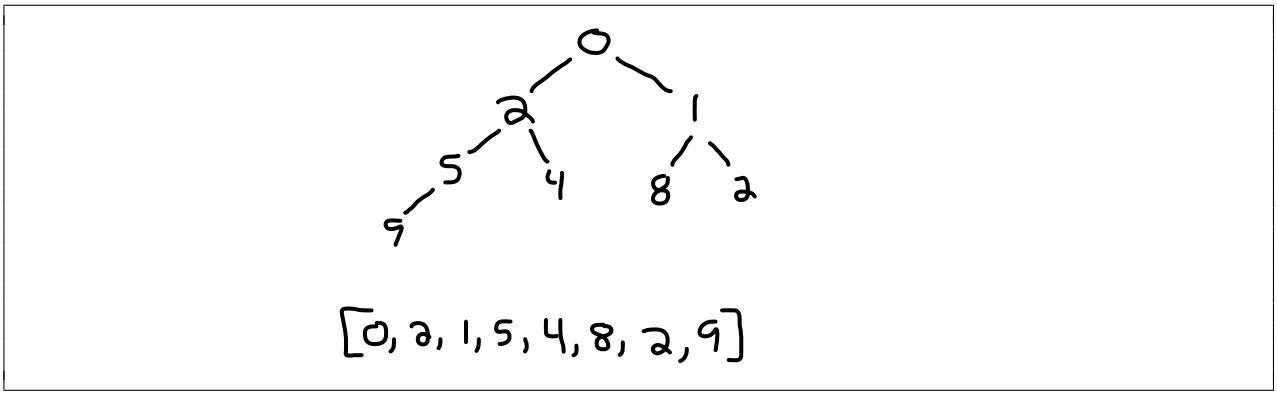


(d) [5 points] Complete, k = 2, height = 4

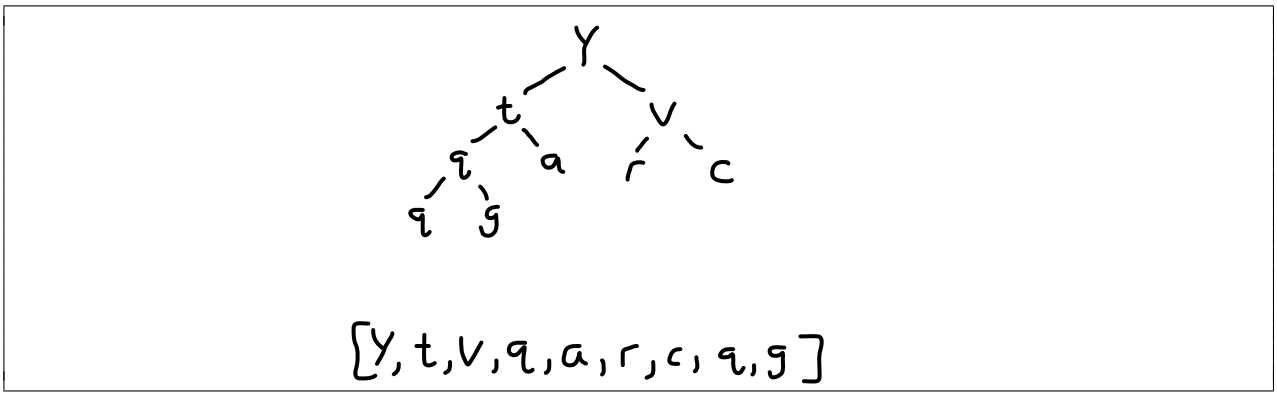


14. [15 points] Take the following sequences and apply the buildHeap algorithm, draw the tree *and* the array which represent the final state of the heap after the algorithm finishes. Strings and characters are in lexicographic order.

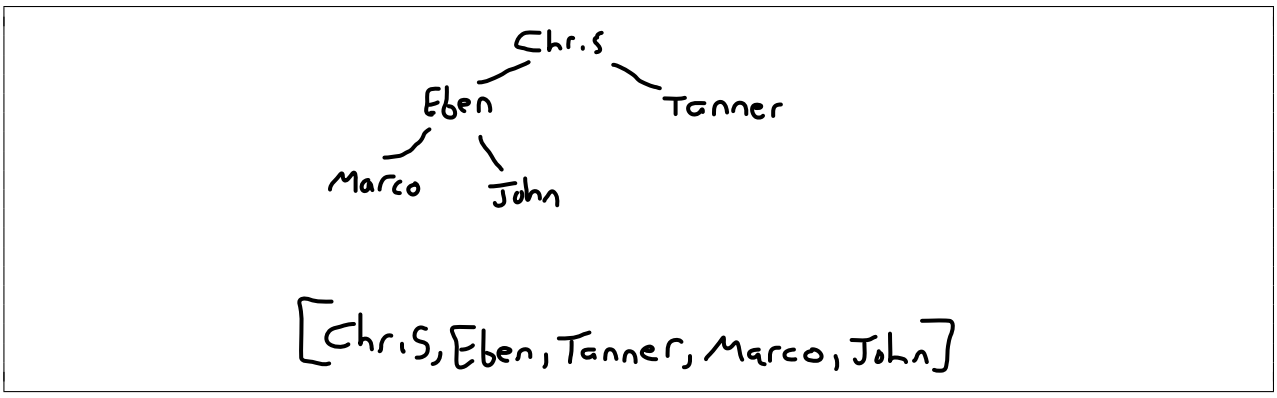
    (a) [5 points] Min Heap: [0, 4, 2, 5, 2, 8, 1, 9]



    (b) [5 points] Max Heap: [a, g, r, q, t, v, c, y, q]

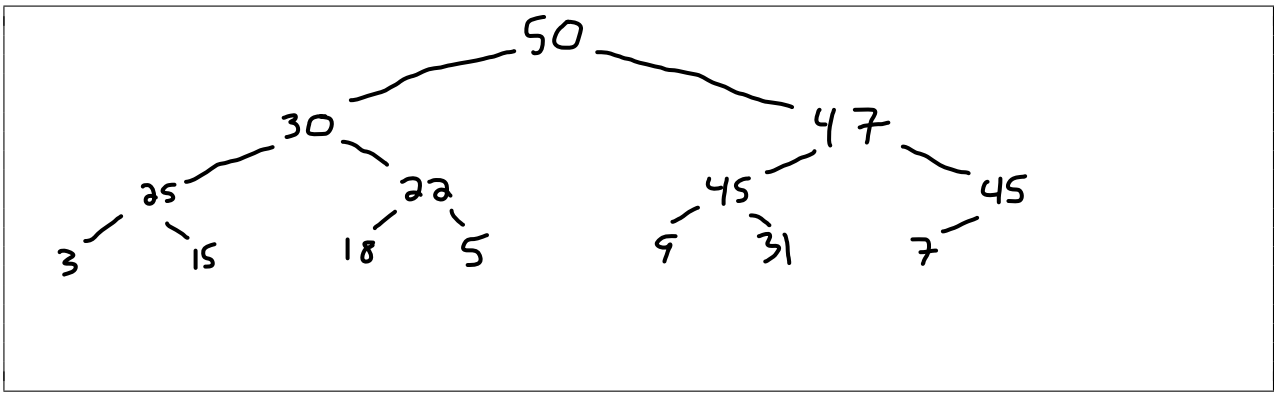


    (c) [5 points] Min Heap: ["Marco", "John", "Tanner", "Eben", "Chris"]

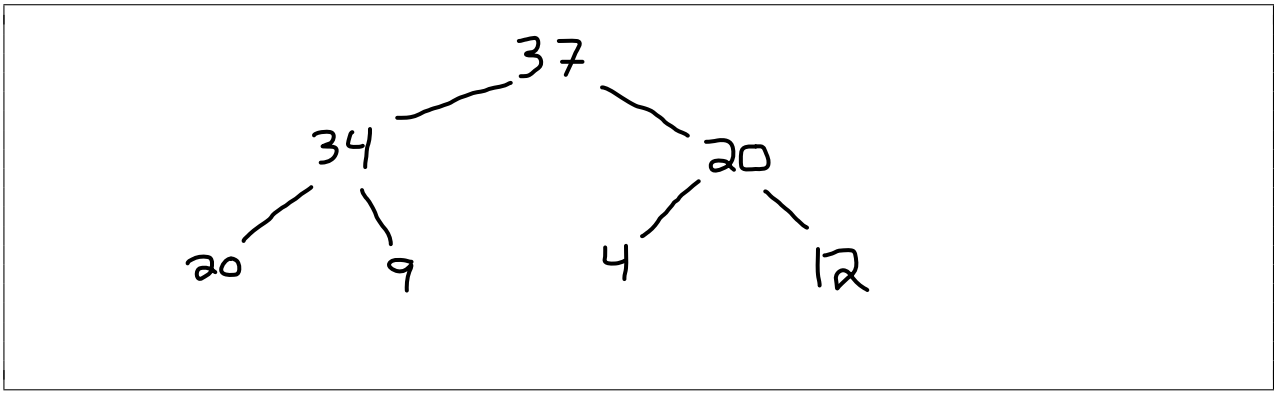


15. [20 points] Take the following sequences representing min and max heaps and draw the resulting heap after the operation.

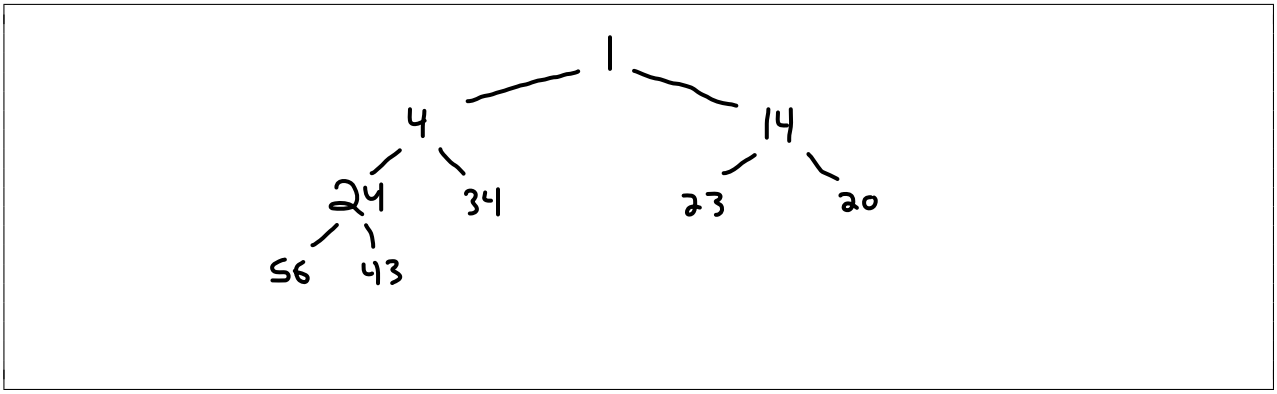
(a) [5 points] Insert 47 into max heap [50, 30, 45, 25, 22, 45, 7, 3, 15, 18, 5, 9, 31]



(b) [5 points] Remove max value from max heap [43, 37, 20, 20, 34, 4, 12, 9]



(c) [5 points] Insert 4 into min heap [1, 24, 14, 43, 34, 23, 20, 56]



(d) [5 points] Remove min value from min heap [4, 23, 12, 23, 23, 76, 23, 79, 95, 79, 45, 80]