- abaacabaab
- acabaacaca
- abbabbaabb
- baababababaa
- cbccbacbcb

## Question 2

**2 pts**

Five (5) elements are added to a mystery data structure. The elements are added in this order: elementA first, elementB second, elementC third, elementD fourth, and elementE last. After calling the remove method TWICE on the mystery data structure, the only three elements that remain are elementA, elementC, and elementE. What could the mystery data structure be?

- Deque
- Stack
- Binary Heap
- Queue

## Question 3

**2 pts**

The file system on your computer is organized into files (documents, images, applications, etc.) and folders (containers for files & folders). To make navigation of these folders intuitive, most computers organize files into a hierarchy starting from a root directory (or folder) which splits important files into sub-folders. Then, these folders can be empty or contain any amount of files or other folders, and so on. What structure is best suited to implement a file system?

- Stack
- Binary Tree
- 2-4 Tree
- Tree
- LinkedList

## Question 4

2 pts

You are a bioinformatics intern working on analyzing DNA sequences for your lab. DNA sequences are represented by strings of characters representing nucleobases. Each nucleobase is represented by a single, unique alphabetical character. The number of possible unique characters is fairly small at only 4 types of nucleobases. You are tasked with identifying the occurrences of a small, specific DNA sequence among hundreds of sample sequences that are each 1000+ characters long. What pattern matching algorithm is best suited to perform this task?

- Quickselect
- KMP
- Boyer Moore
- Dijkstra's
- Brute Force

## Question 5

2 pts

Starting from an empty HashMap with backing array of **length 8** that uses **linear probing** for collision resolution, perform the operations below and choose the resulting backing array. The elements below are keys, and their corresponding values are omitted. The hash function is the key itself. The load factor is 1.0.

1. Add 2
2. Add 4
3. Add 10
4. Add 11
5. Remove 4
6. Add 18

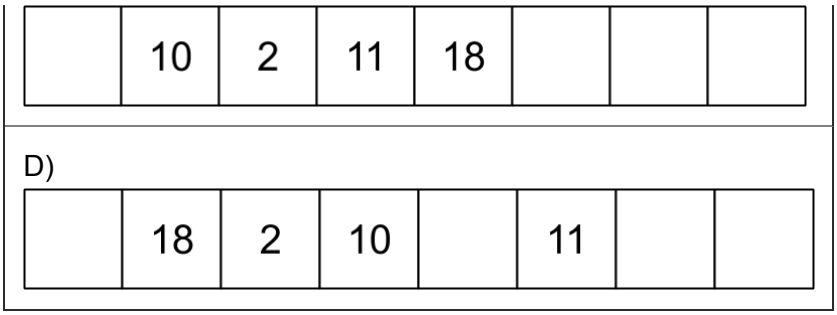
A)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 2 | 10 | 18 | 11 | | |

B)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 2 | 10 | | 11 | 18 | |

C)

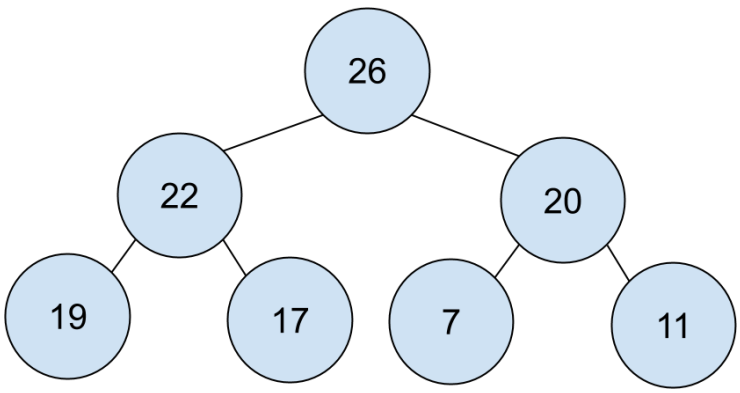



- C
- B
- A
- D

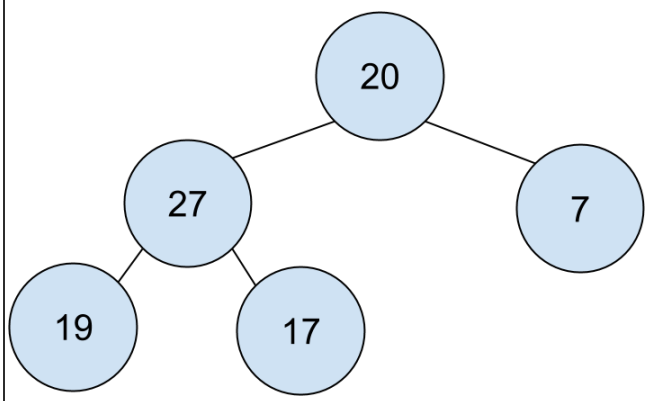
## Question 6

2 pts

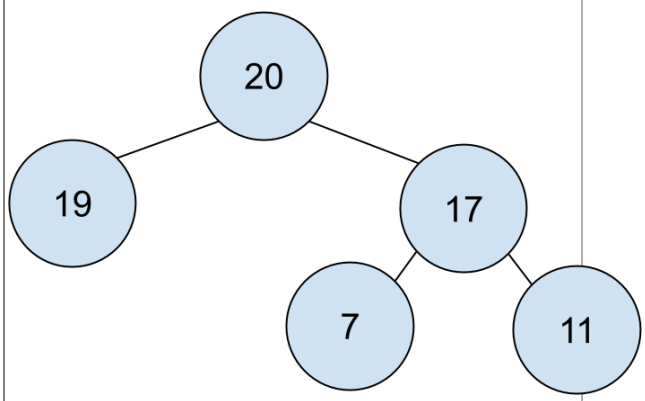
Given the following initial **Max** Heap, perform two removals and choose the resulting Heap from the options in the table.




A)

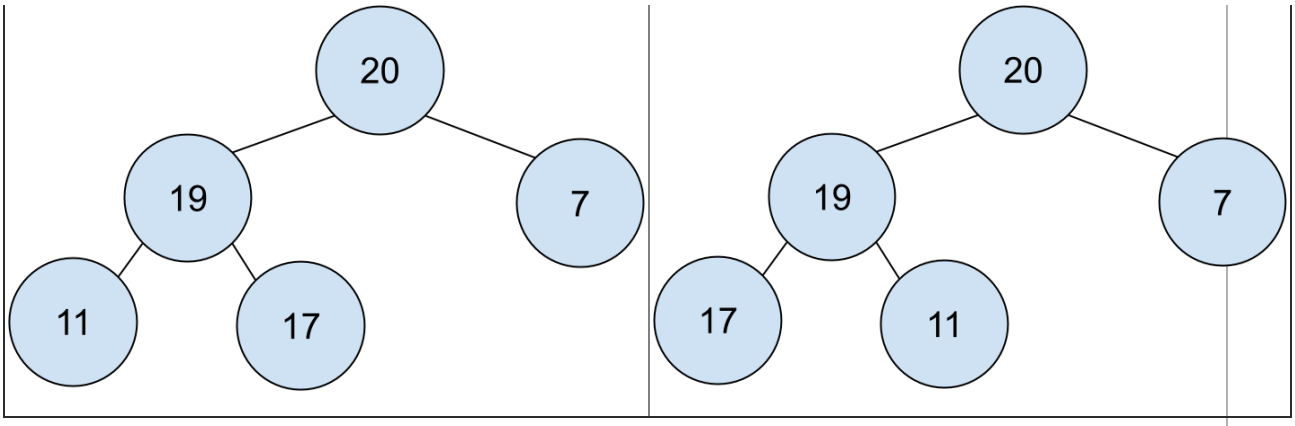



B)




C)

D)



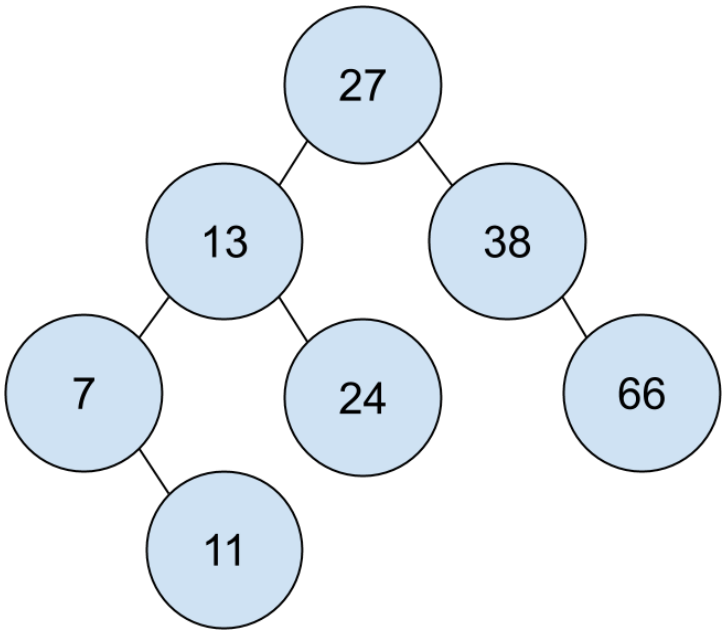

- B
- C
- D
- A

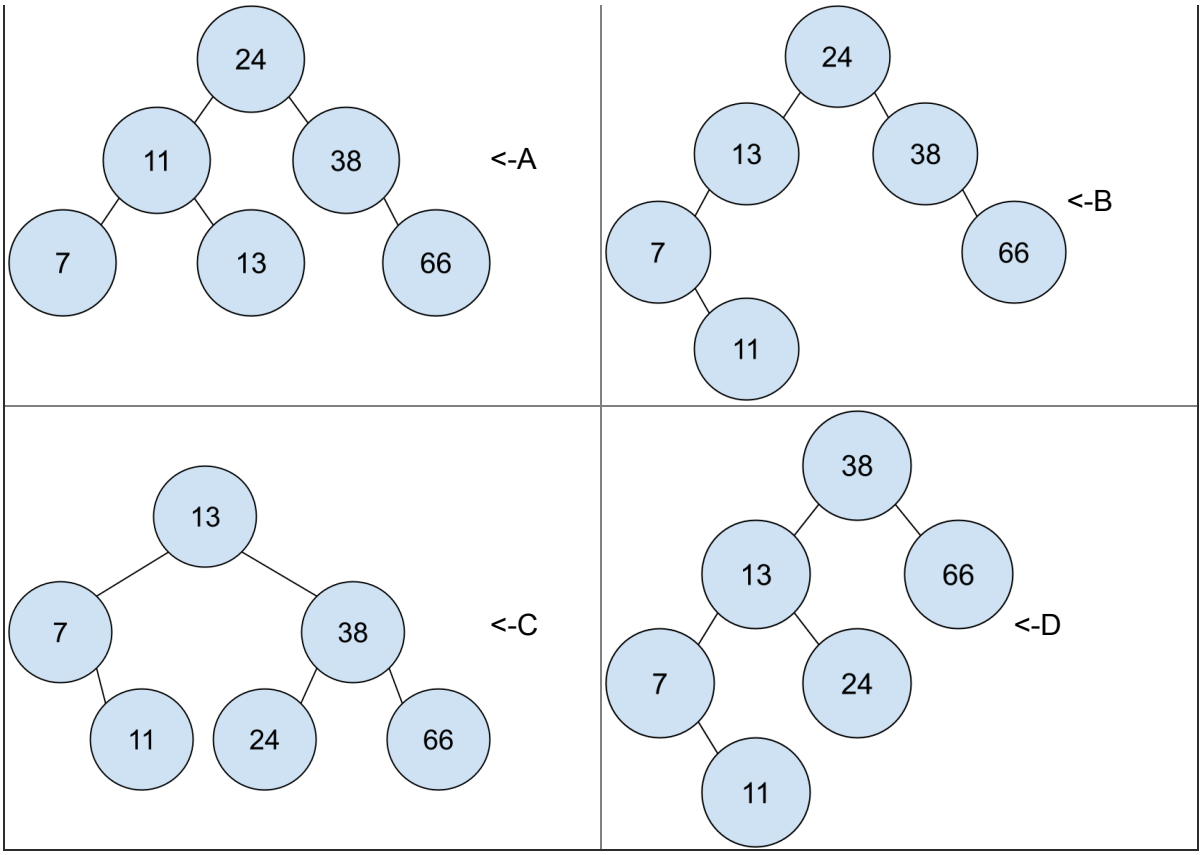
## Question 7

2 pts

What is the result of removing **27** from the AVL displayed below? (Use the predecessor, if necessary)

Starting AVL:






- C
- B
- A
- D

## Question 8

2 pts

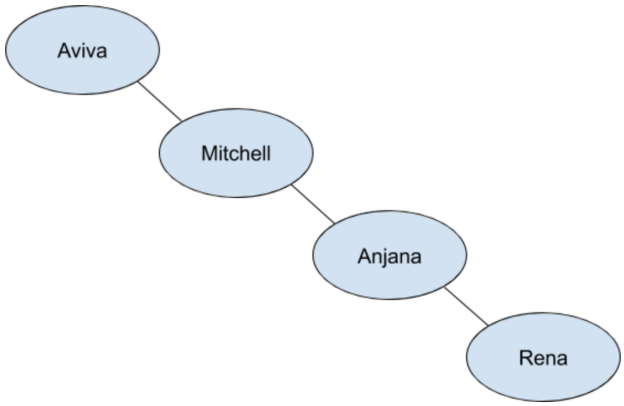
Which of the following statements about Doubly Linked Lists and Singly Linked Lists is **TRUE**?

- The time complexity of removing the middle node (i.e. the node at index n/2) from both a Doubly Linked List and a Singly Linked List of the same length is the same.
- It is faster to add to the front of a Doubly Linked List than a Singly Linked List.
- When they each have the same number of elements, a Singly Linked List will take up more space in memory than a Doubly Linked List.
- When both have a tail pointer, it is faster to remove at the back of a Singly Linked List than a Doubly Linked List.
- None of these choices.

## Question 9

2 pts

You are given the following binary tree and a traversal output of the tree. **Choose ALL traversals that could produce the output**. If none apply, then leave all check boxes empty.



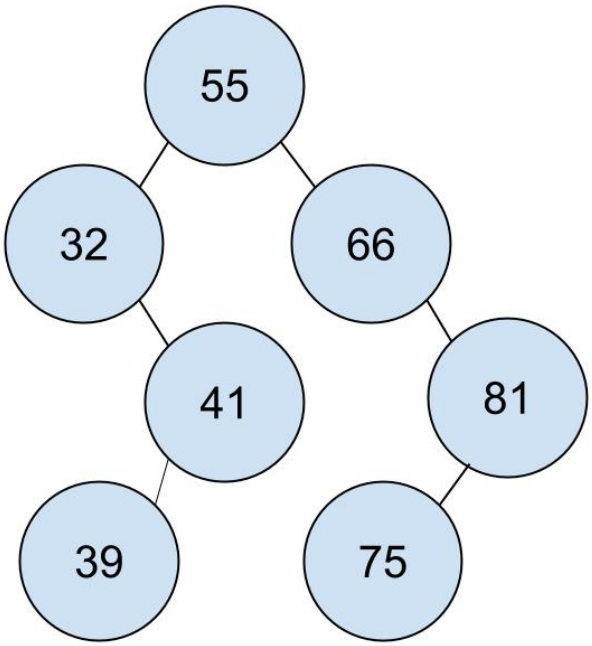

**Output:** [Aviva, Mitchell, Anjana, Rena]

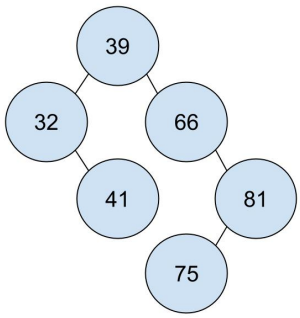
- [ ] Inorder
- [ ] Levelorder
- [ ] Preorder
- [ ] Postorder
- [ ] KMP

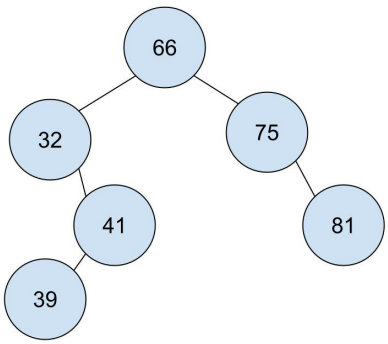
## Question 10

2 pts

What is the result of removing **55** from the BST displayed below? If necessary, use the **predecessor**.

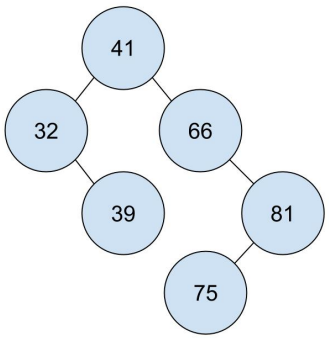


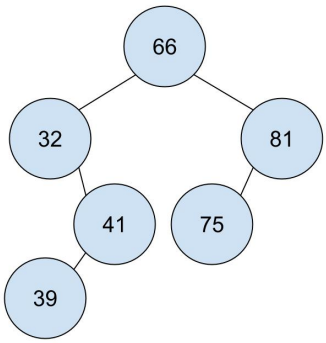




<- A

B ->




<- C




<- D

- A
- B
- D
- C

## Question 11

2 pts

Perform **2** complete iterations of LSD Radix sort on the initial array shown below. Then select the choice that matches the resulting array. The first row displays the indices of the array, the second (bottom) row displays the actual array elements that are being sorted.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 6177 | 6471 | 1771 | 6715 | 3635 | 2415 | 6905 | 4240 |

- | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
  |---|---|---|---|---|---|---|---|
  | 4240 | 1771 | 6471 | 3635 | 6715 | 6905 | 2415 | 6177 |
- | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
  |---|---|---|---|---|---|---|---|
  | 6905 | 2415 | 6715 | 3635 | 4240 | 1771 | 6471 | 6177 |
- | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
  |---|---|---|---|---|---|---|---|
  | 4240 | 6471 | 1771 | 6715 | 3635 | 2415 | 6905 | 6177 |
- | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
  |---|---|---|---|---|---|---|---|
  | 6905 | 6715 | 2415 | 3635 | 4240 | 6471 | 1771 | 6177 |

## Question 12

2 pts

Which of the following sorting algorithms are stable?

a) Insertion Sort

b) Quick Sort

c) Bubble Sort

d) LSD Radix Sort

e) Selection Sort

- [ ] Insertion Sort
- [ ] Selection Sort
- [ ] Quick Sort
- [ ] Bubble Sort
- [ ] LSD Radix Sort

## Question 13

2 pts

What will the array below look like after **TWO** iterations of Insertion Sort are performed?

```
Initial Array: [110, 38, 102, 63, 44, 56, 101, 43]
```

- ( ) `[38, 110, 102, 63, 44, 56, 101, 43]`
- ( ) `[38, 44, 56, 63, 102, 110, 101, 43]`
- ( ) `[38, 63, 102, 110, 44, 56, 101, 43]`
- ( ) `[38, 63, 44, 56, 101, 43, 102, 110]`
- ( ) None of these choices.

## Question 14

2 pts

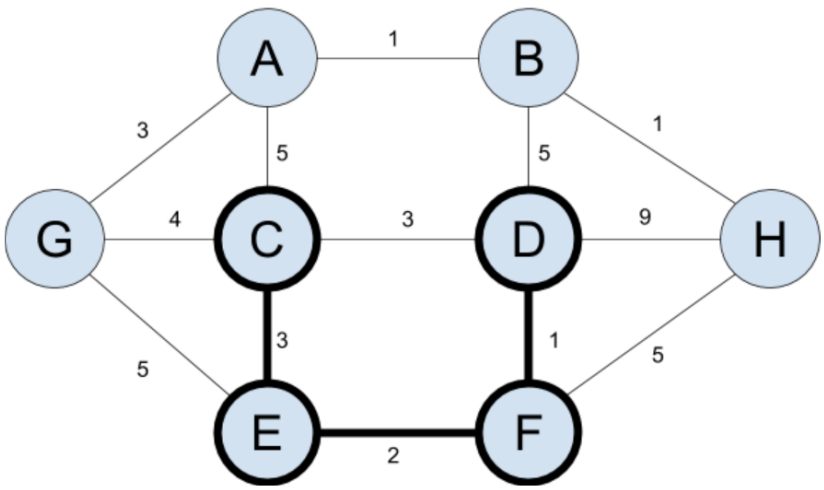
Which of the following is **FALSE** about 2-4 trees?

- A node with 3 data elements must have 4 children.
- Promotion is the 2-4 tree balancing strategy used only during a 2-4 tree add operation.
- None of these choices.
- Transfer, aka. rotation, operations are used only during 2-4 tree remove operations.
- After a transfer (aka. rotation) operation to balance the tree, at least one fusion operation must occur to ensure that the final tree is balanced.

## Question 15

2 pts

You are trying to find an MST for the graph below, so you decide to run Prim's Algorithm. Highlighted in the graph below are the edges that have already been added to the MST at this point in the algorithm. Which of the following edges would be added to the MST next if you continue with the algorithm?




- AB
- CA
- FH
- GC
- CD

## Question 16

2 pts

For the algorithm listed below, determine the time complexity of the implementation, and choose from the selections provided. Make sure you choose the tightest Big-O upper bound possible for the operation.

When performing the **Breadth First Search** algorithm on a sparse, connected graph with |V| vertices and |E| edges that has an adjacency list implementation, what is the **worst case** time complexity of the algorithm?

- O(|E|^2)
- O(|E|)
- O(|V|^2)
- O(|V| + |E|)
- O(|V|)

## Question 17

2 pts

For the algorithm listed below, determine the time complexity of the implementation, and choose from the selections provided. Make sure you choose the tightest Big-O upper bound possible for the operation.

When performing the **Depth First Search** algorithm on a dense graph with |V| vertices and |E| edges that has an adjacency list implementation, what is the **worst case** time complexity of the algorithm?

- O(|E|)
- O(|V|^2)
- O(|V| + |E|)
- O(|E|^2)
- O(|V|)

## Question 18

2 pts

For the algorithm listed below, determine the time complexity of the implementation, and choose from the selections provided. Make sure you choose the tightest Big-O upper bound possible for the operation.

What is the time complexity of running **Dijkstra's Shortest Path** algorithm on a graph with |V| vertices and |E| edges and an adjacency list representation?

- O((|E| + |V|)log|V|)
- O(|E| + |V|)
- O(|V| + |E|log|V|)
- O( |V| log |V| )
- O(|E| + |V|log|V|)

## Question 19

**2 pts**

For the algorithm listed below, determine the time complexity of the operation, and choose from the selections provided. Make sure you choose the tightest Big-O upper bound possible for the operation.

When performing the Boyer Moore algorithm on text of length *n* and a pattern of length *m* where every character in the text is not in the alphabet of the pattern, what is the worst case time complexity of the algorithm?

- O(mn)
- O(m + n)
- O(m + n/m)
- O(n)
- O(m)

## Question 20

**2 pts**

For the algorithm listed below, determine the time complexity of the operation, and choose from the selections provided. Make sure you choose the tightest Big-O upper bound possible for the operation.

When performing the KMP algorithm on text of length *n* and a pattern of length *m* where every character in the pattern does not repeat, what is the worst case time complexity of the

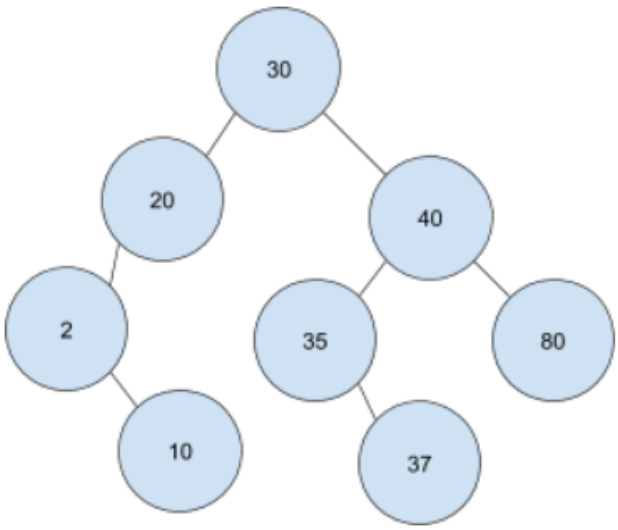
algorithm?

- O(mn)
- O(m)
- O(m + n/m)
- O(n)
- O(m + n)

## Question 21

2 pts

Given the following tree:



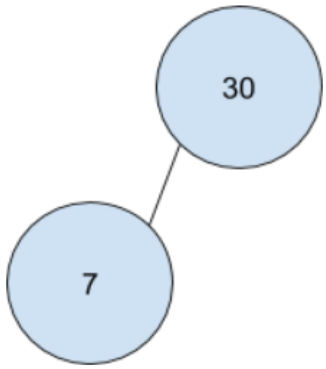

Select **all of** the options below that describe the tree.

- [ ] BST
- [ ] Heap
- [ ] AVL
- [ ] Binary Tree

## Question 22

2 pts

Given the following tree:



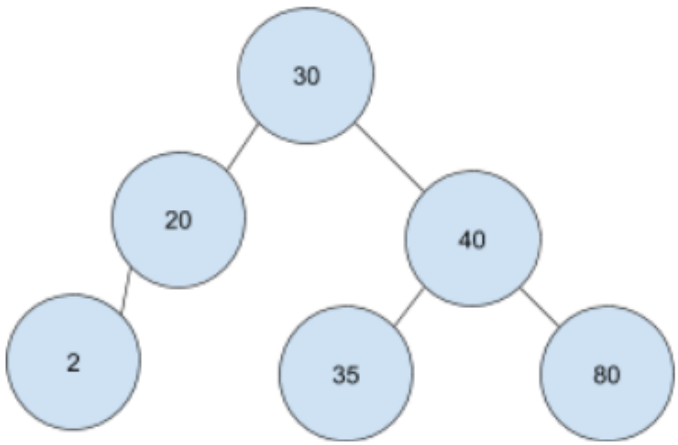

Select **all of** the options below that describe the tree.

- [ ] AVL
- [ ] Heap
- [ ] BST
- [ ] Binary Tree

## Question 23

**2 pts**

Given the following tree:



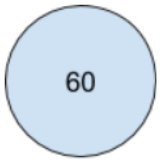

Select **all of** the options below that describe the tree.

- [ ] Heap
- [ ] Binary Tree
- [ ] BST
- [ ] AVL

## Question 24

2 pts

Given the following tree:




Select **all of** the options below that describe the tree.

- [ ] AVL
- [ ] Binary Tree
- [ ] BST
- [ ] Heap

## Question 25

2 pts

The following array **has had 2 full iterations of Cocktail Shaker Sort** performed on it. The first row displays the indices of the array, the second (bottom) row displays the actual array elements that have been partially sorted.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 26 | 36 | 62 | 38 | 59 | 64 | 87 |

Select from the below options the **original array before 2 full iterations of Cocktail Shaker Sort** were performed.

- ○

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 38 | 26 | 87 | 64 | 36 | 62 | 59 |

- ○

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 38 | 26 | 87 | 36 | 64 | 62 | 59 |

- ○

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| 64 | 87 | 62 | 38 | 36 | 26 | 59 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 59 | 62 | 38 | 26 | 36 | 87 | 64 |

## Question 26

1 pts

For at least one TA that was particularly helpful to you this semester, write the name of a song or write them an original song. If you write an original song, do not write explicit lyrics or you will not get credit. Original songs that surpass the expectations of the TAs are eligible to receive **an additional point**. You can write the name of the TA(s) your song is for in the entry box below, and write the name of the song under the TA name(s). The final exam is **not** an appropriate medium for making **advances** or **inappropriate comments** towards a teaching assistant. Inappropriate submissions will earn a **0** for the entire exam.

**Head TA** - Adrianna Brown

**Senior TA/B3 Grading -** David Wang

**Senior TA/B3 Grading -** Rodrigo Pontes

**Online Head TA -** Caroline Kish

**O1 -** Jacob Allen

**O1 -** Landon Ryan

**O1 -** Isaac Weintraub

**A1 -** Paige Ryan

**A1 -** Tillson Galloway

**A2 -** Yotam Kanny

**A2 -** Aviva Kern

**A3 -** Destini Deinde-Smith

**A3 -** Siddu Dussa

**A4/A5/A6/GR1 -** Neha Deshpande

**A4/A5/A6/GR1 -** Isaac Tomblin

**B1 -** Reece Gao

**B1 -** Rena Li

**B2 -** Miguel de los Reyes

**B2 -** Smita Mohindra

**B3 -** Sanjana Tewathia

**B4 -** Mitchell Gacuzana

**B4 -** Eunseo Cho

**B5 -** Elena May

**B5 -** Ivan Leung

**B6 -** Anjana Nandagopal

**B6 -** Alex McQuilkin

**C1 -** Brooke Miller

**C1 -** Cliff Panos

**C2/GR2 -** Brandon Vu

**C2/GR2 -** Ila Vienneau

**C3/C4 -** Nick Worthington

**C3/C4 -** Keely Culbertson

HTML Editor

12pt Paragraph

0 words

Quiz saved at 4:08pm Submit Quiz