Question 1

4 pts

Given an array which contain five integers, int arr[] = {19, 7, 2, 11, 6}. How many swaps are performed by the Bubble Sort algorithm, if we sort the array in ascending order?

Given an array of 10 numbers, int a[] = {99, 27, 14, 2, 7, 14, 17, 5, 1, 44}. Let's use the selection sort algorithm, to sort the array in ascending order. If we inspect the contents of our array, when i == 3, what value will be stored at index four in the array after the swap was performed.

For your reference, here is the selection sort algorithm that was obtained from its Wikipedia page. Use this to trace the logic and determine what value will be in a[4]

```
/* a[0] to a[aLength-1] is the array to sort */
int a[] = {99, 27, 14, 2, 7, 14, 17, 5, 1, 44};
int i,j;
int aLength = 10; // initialize to a's length

/* advance the position through the entire array */
/*   (could do i < aLength-1 because single element
is also min element) */
for (i = 0; i < aLength-1; i++)
{
    /* find the min element in the unsorted a[i .. a
Length-1] */

    /* assume the min is the first element */
    int jMin = i;
    /* test against elements after i to find the sma
llest */
    for (j = i+1; j < aLength; j++)
    {
        /* if this element is less, then it is the n
ew minimum */
        if (a[j] < a[jMin])
        {
            /* found new minimum; remember its index
*/
            jMin = j;
        }
```

```
/* a[0] to a[aLength-1] is the array to sort */
int a[] = {99, 27, 14, 2, 7, 14, 17, 5, 1, 44};
int i,j;
int aLength = 10; // initialize to a's length

/* advance the position through the entire array */
/*   (could do i < aLength-1 because single element
is also min element) */
for (i = 0; i < aLength-1; i++)
{
    /* find the min element in the unsorted a[i .. a
Length-1] */

    /* assume the min is the first element */
    int jMin = i;
    /* test against elements after i to find the sma
llest */
    for (j = i+1; j < aLength; j++)
    {
        /* if this element is less, then it is the n
ew minimum */
        if (a[j] < a[jMin])
        {
            /* found new minimum; remember its index
*/
            jMin = j;
        }
    }

    if (jMin != i)
    {
        swap(a[i], a[jMin]);
    }
//TODO: if i == 3, what value will be stored in a[4]
at this point in the code.
}
```

Which of the following algorithms are stable? Select all that apply.

- [ ] Bubble Sort
- [ ] Mergesort
- [ ] Heapsort
- [ ] Insertion Sort
- [ ] Quick Sort

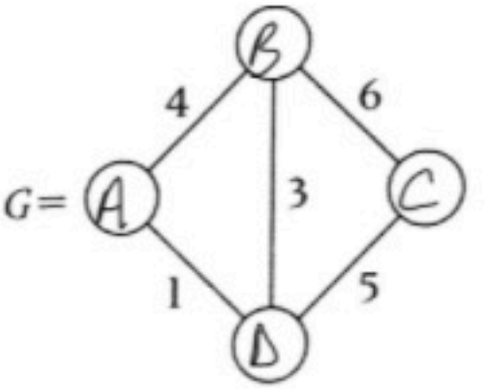
Question 4

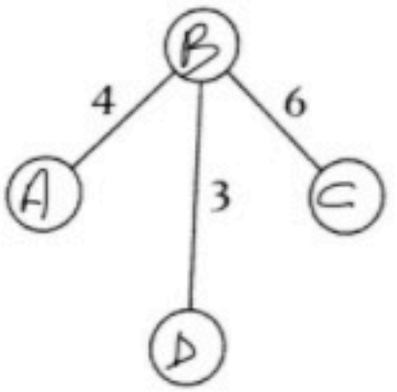
5 pts

Match each algorithm with its respective worst case time complexity.

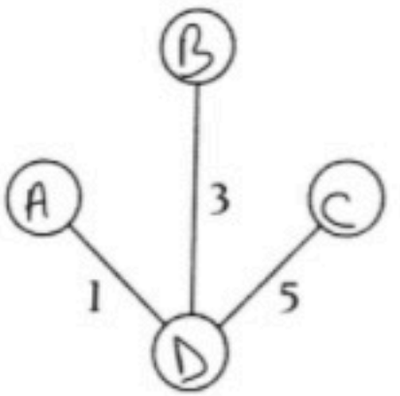
Insertion Sort	O(n^2)
Quick Sort	O(n^2)
Bubble Sort	O(n^2)
Merge Sort	O(n log n)
Heap Sort	O(n log n)
Selection Sort	O(n^2)
Binary Search Tree	O(n)

For the given graph, G, which of the following trees (labeled from a - h) are possible spanning trees. Select all that apply.

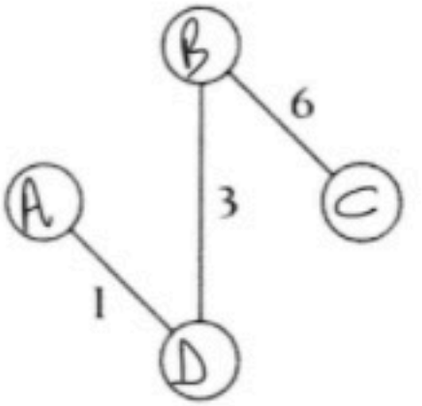




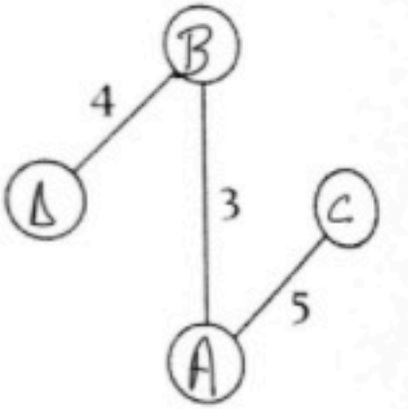
(a)



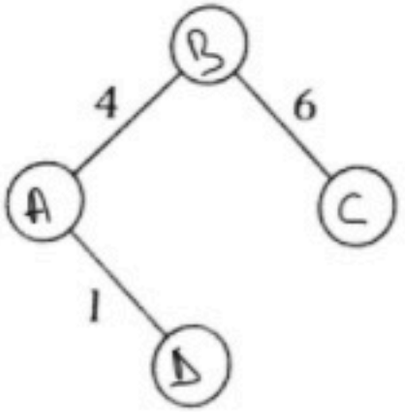
(b)



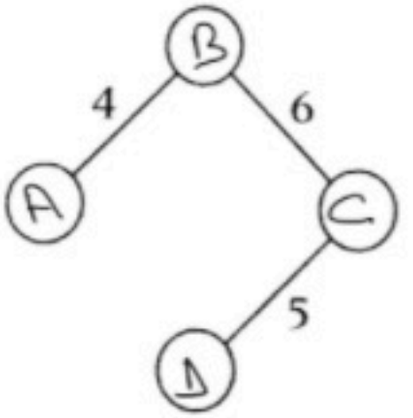
(c)



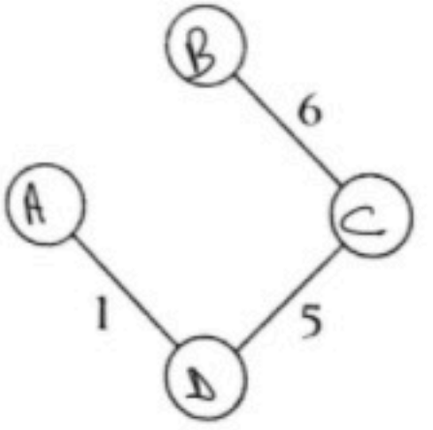
(d)



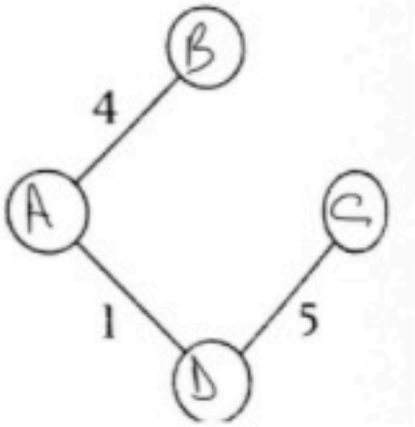
(e)



(f)



(g)



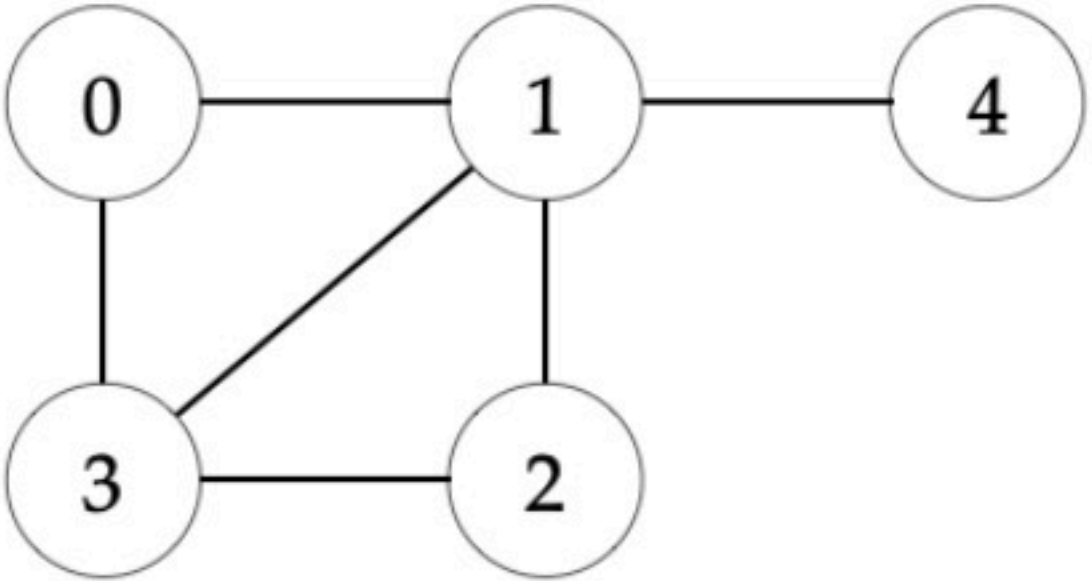
(h)

- [ ] a
- [ ] f
- [ ] b
- [ ] None of the above
- [ ] d
- [ ] h
- [ ] e
- [ ] g
- [x] All of the above
- [ ] c

Question 6

5 pts

Given the graph below, which of the following is its correct adjacency matrix:



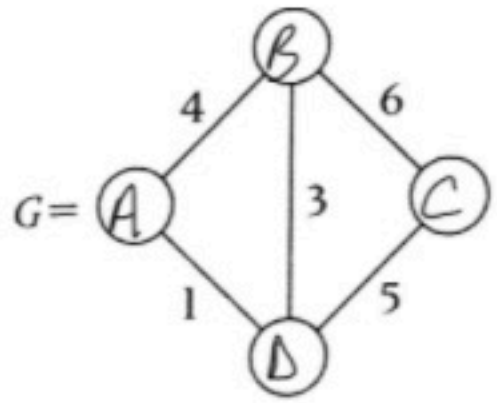
	0	1	2	3	4
0	0	1	0	1	0
1	1	0	1	0	1
2	0	0	0	1	0
3	1	0	1	0	0
4	0	1	0	0	1

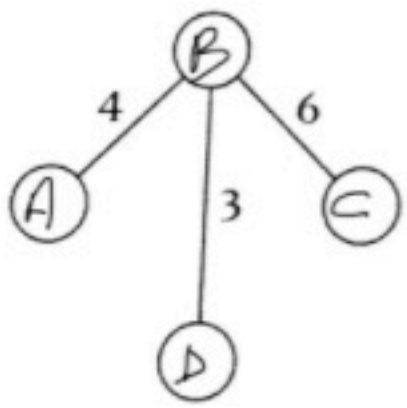
	0	1	2	3	4
0	0	1	0	1	0
1	1	0	1	1	1
2	0	1	0	1	0
3	1	1	1	0	0
4	0	1	0	0	0

Question 7

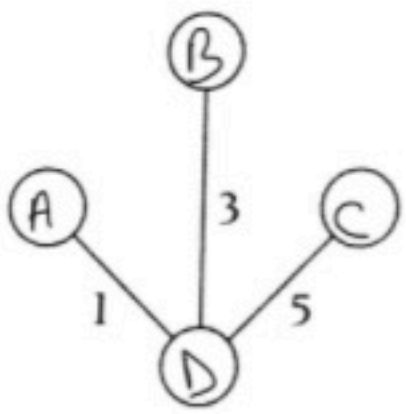
5 pts

For the following graph, G, and its possible spanning trees labeled (a) to (h). What is the cost of the minimum spanning tree?

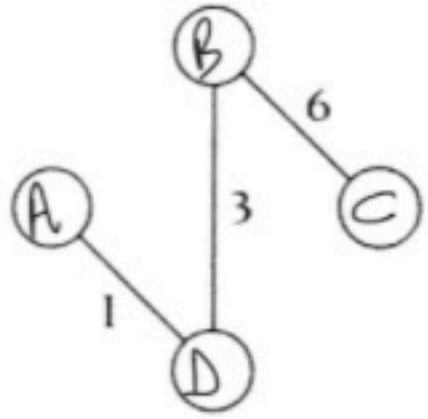




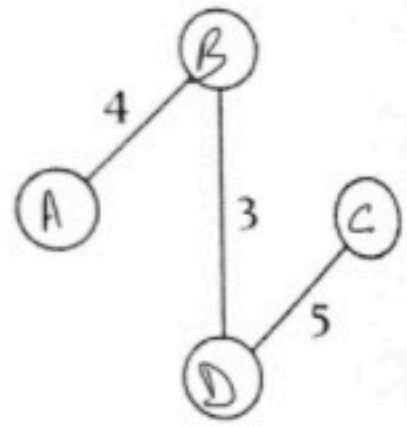
(a)



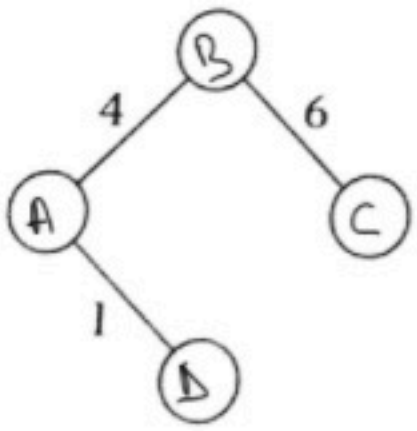
(b)



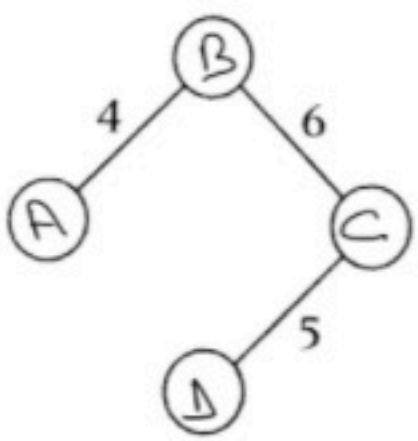
(c)



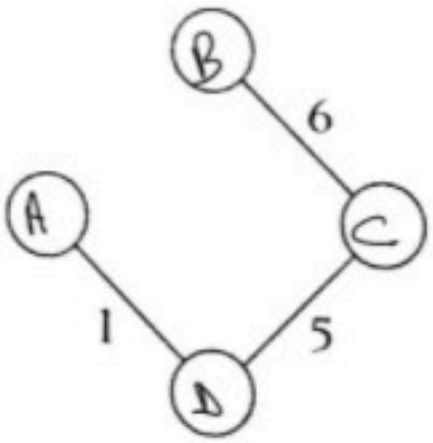
(d)



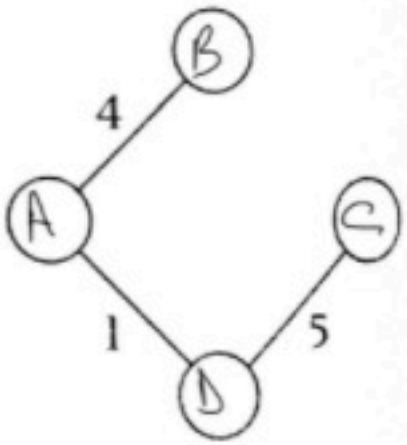
(e)



(f)



(g)

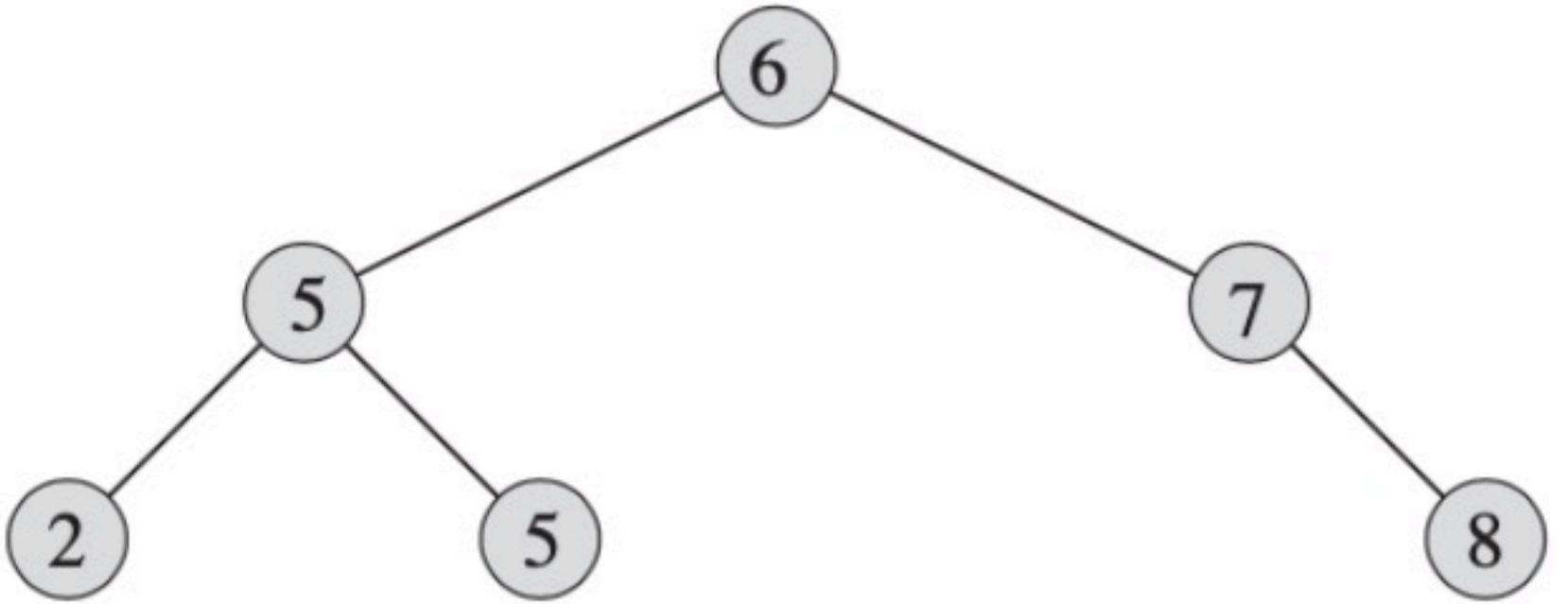


(h)

Question 8

3 pts

Given the following binary search tree, what is the resulting sequence if a *pre-order traversal* is performed? Ensure that your response is comma separated, e.g. 1,2,3,4

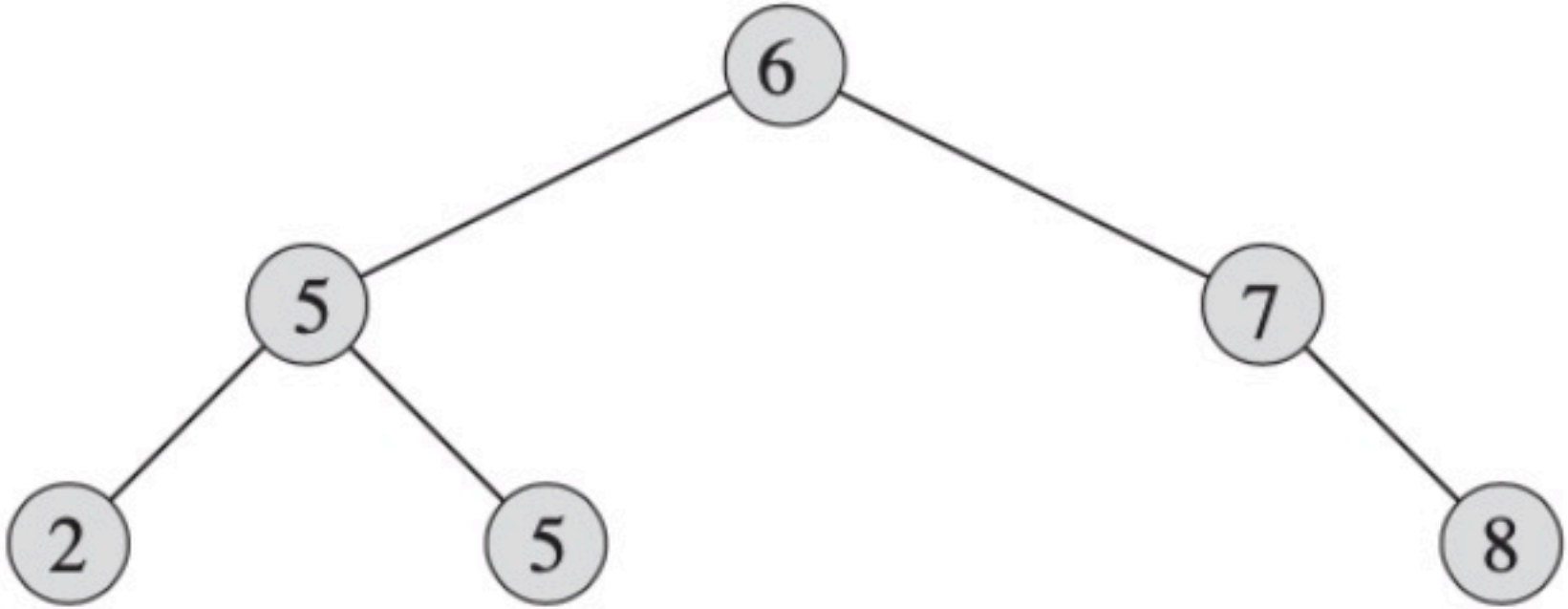


6,5,2,5,7,8

Question 9

3 pts

Given the following binary search tree, what is the resulting sequence if a *breadth-first search traversal* is performed? Ensure that your response is comma separated, e.g. 1,2,3,4

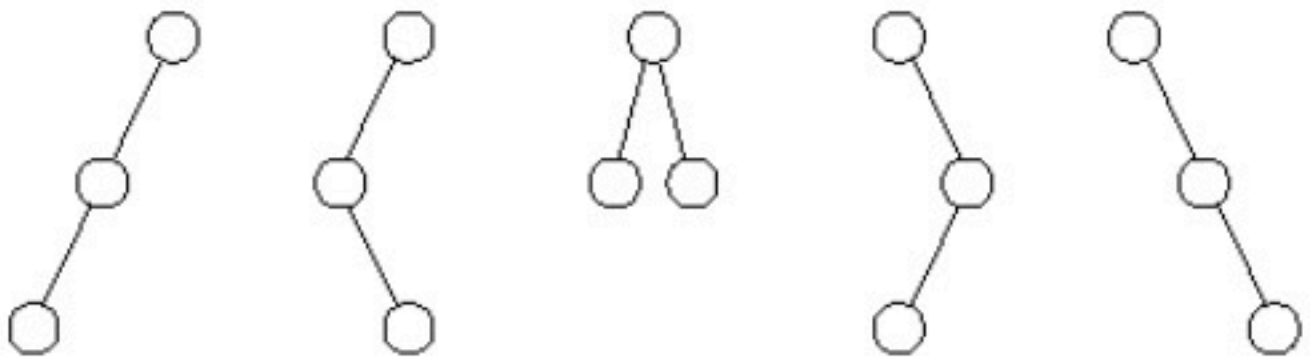


6,5,7,2,5,8

Question 10

3 pts

Given the diagram below, indicate which of the following is a Binary Tree? Select all that apply.



(a) (b) (c) (d) (e)

- [x] c
- [x] b
- [x] e
- [x] d
- [x] a

Question 11

2 pt

For any given graph, G, there can only be one (1) minimum spanning tree.

- ○ True
- ○ False

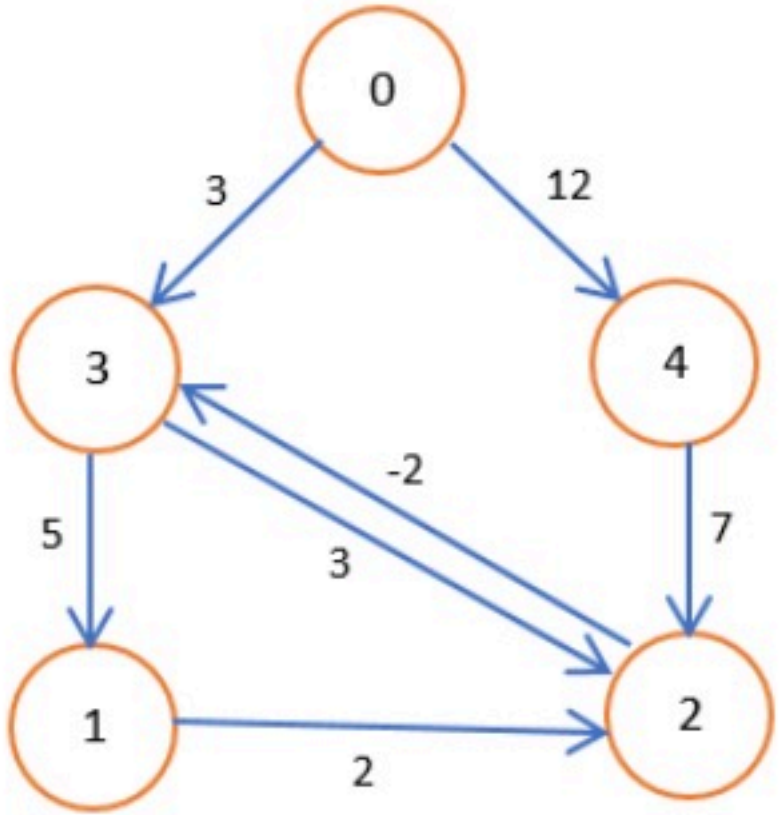
Question 12

2 pts

Greedy algorithms always find the optimal solution for optimization problems.

- ○ True
- ● False

Does the following graph contain any cycles?



- ○ True
- ○ False

Question 15

2 pts

Memoization facilitates a computational tradeoff between time and space complexity?

- (●) True
- ( ) False

Question 16

2 pts

When can you use binary search and still have an $O(\log n)$ runtime?

- ○ with a sorted doubly-linked list
- ● with a sorted array
- ○ with an unsorted array
- ○ with an unsorted doubly-linked list

Question 17

Deleting from a hashtable is ...

- ○ $O(\log n)$ $O(\log n)$
- ● $O(1)$ $O(1)$
- ○ depends on whether it is sorted
- ○ $O(n)$

Question 18

Which of the following best describes *asymptotic bounding*?

- ○ Represents the best case runtime of an algorithm
- ○ Represents an approximation of runtime that is independent of input size
- ○ Is a function that classifies the runtime family than an algorithm belongs to.
- ○ Represents an target runtime that we do not want an algorithm to exceed

Question 19

The RAM model of computation ...

- ○ helps us reason about algorithmic memory requirements
- ○ helps us reason about algorithmic performance
- ○ helps us reason about algorithmic correctness
- ○ helps us reason about the caching effects of an algorithm

Question 20

2 pts

Which of the following best applies to an algorithm?

- (●) Is a set of steps to be followed to solve a problem
- ( ) Requires a computer to run
- ( ) All of the others.
- ( ) Should be run in an thread.

Question 21

2 pts

Which of the following is not an example of a big-Oh runtime?

- ● Polynomial
- ○ Quadratic
- ○ Linear
- ○ Constant

Question 22

2

Which of the following relates to implementing recursion in C?

- ○ can be optimized into a loop when written in a tail-recursive way
- ○ all of the others.
- ○ adds to the execution (or call) stack until the base case is reached
- ○ is an alternative way to implement a loop

Question 23

Which of the following is true about merge sort?

- [ ] merges the elements around a pivot
- [x] always results in a tree that has $\log_2 n$ levels
- [ ] can be implemented *in place*
- [x] does $O(n)$ work at each level in the tree

Question 24

2 pts

Which of the following is an advantage that insertion sort has over merge sort?

- ○ None of the others
- ○ It has a better runtime
- ○ It only has to swap element once
- ● It can be done *in place*

Question 25

2 pts

There are many sorting algorithms because ...

- ○ the best sorting algorithm depends on the number of swaps it must perform
- ○ the best sorting algorithm depends on how easy it is to implement
- ● the best sorting algorithm depends on the nature of the problem
- ○ the best sorting algorithm depends on the size of the input

The add function for trees ...

- ○ All of the others
- ○ always adds a leaf node
- ○ is O(1)
- ○ first checks the root to see if it is null

Which of the following are characteristics of a *heap*?

- [ ] a binary tree
- [ ] the value in the parent is greater than the value in all of its children
- [ ] is complete
- [ ] value of left child < value of parent < value of right child

Question 27

2 pts

Which of the following are characteristics of a *heap*?

- [x] a binary tree
- [ ] the value in the parent is greater than the value in all of its children
- [x] is complete
- [ ] value of left child < value of parent < value of right child

Which of the following *best* describes the Tree data structure?

- ○ Trees are very similar to a doubly linked list because it has two pointers to other nodes
- ○ Trees represent a more complicated way of storing data
- ○ Trees represent a data hierarchy
- ○ Trees are good for storing data sequentially

Question 29

2 pts

A *breadth-first search* traversal ...

- ○ has a time complexity that is bounded by the height of the tree
- ● visits each node while keeping track of levels
- ○ is implemented recursively
- ○ starts at the leaves and moves up through the tree

Question 30

2 pts

Which of the following is a property of a *complete* tree?

- ○ last level has all left children
- ○ leaf nodes are all are all on the last level
- ● last level has all nodes to the left
- ○ all levels, except the last one, are not full

Question 31

2 pts

Which of the following is true about nodes in a *binary tree*?

- [ ] can have up to 2 children
- [ ] can have 0 or 2 children
- [ ] can have up to 2 parents
- [ ] can have up to 2 siblings

Question 32

2 pts

Why do we need to keep track of which nodes are visited (or discovered) when we are implementing a graph traversal?

- (●) to detect when a graph has cycles
- ( ) to ensure that we do traverse each node exactly one time
- ( ) to avoid implementing the traversal recursively
- ( ) to make the algorithm different from traversing a tree

Question 33

2 pts

Breadth-first traversals of a graph

- ○ can be implemented recursively to avoid using a stack
- ● use a queue to store the vertices that have not been traversed
- ○ use a queue to store the vertices that have been traversed
- ○ use a stack to store the vertices that have not been traversed

Question 34

2 pts

Which of the following best describes a *greedy algorithm*?

- ○ Work by always choose the "best" option at each step
- ○ Similar to divide and conquer, they take steps towards making the problem smaller
- ● All of the others
- ○ Always generate a unique, correct solution

Question 35

2 pts

Which of the following graphs can you apply Dijkstra's algorithm to?

- [ ] undirected graph with negative edge weights
- [ ] directed graph with negative edge weights
- [ ] undirected graph with non-negative edge weights
- [x] directed graph with non-negative edge weights

Which of the following greedy strategies works best for the *interval scheduling problem*?

- ● fewest interruptions
- ○ ends first
- ○ shortest interval
- ○ starts first

Question 37

2 pts

Which of the following best describes when Dijkstra's Algorithm stops?

- ○ when we have completed a DFS of the graph
- ○ when we find the shortest path
- ○ when we have completed a BFS of the graph
- ○ when we have looked at all the edges

Question 38

2 pts

Which of the following best describes the naive recursive Fibonacci implementation?

- ○ requires a lot of memory dynamically allocated memory that must be freed
- ○ requires a lot of recomputation, resulting in linear runtime
- ● requires a lot of recomputation, resulting in exponential runtime
- ○ requires a lot of recursive calls resulting in a large execution stack and exponential space

Question 39

2 pts

Which of the following best describes *dynamic programming* strategy?

- ○ solves the problem using the best possible local choice
- ● divides problem and solves the smaller version
- ○ solves the problem using traversal strategies
- ○ solves the problem using brute force

Question 40

2 pts

Which of the following are we referring to when we talk about the space-time tradeoff?

- (●) time savings
- ( ) both space and time savings
- ( ) space savings
- ( ) depends on the algorithm