Name
Email

Computer Science 22
Practice Final Exam

Marking Instructions

Be sure to completely fill in the appropriate bubble.

Example

● B C D E

	A	B	C	D	E
1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E

This exam consists of three parts:

- Part I has multiple-choice questions that you must complete, darkening the bubble for each answer in the grid shown at left. ***Make sure to completely fill in the appropriate bubble and erase any stray marks.***
- Part II consists of 4 multi-part problems, of which you must complete 3. Each of these problems will have a box that looks like this:

  _____ GRADE _____ DON'T GRADE

  Make sure to clearly fill out the box accompanying each problem to indicate if it should be graded. ***If you leave the boxes blank, we will <u>not</u> count your best 3 problems.***
- Part III consists of a three-part problem that you must complete.

Keep all pages stapled together.

You have **two hours** to complete the exam. The questions are worth a total of 100 points. In order to budget your time, plan on spending approximately one minute per point.

You may use a single 8.5 x 11-inch sheet of notes (handwritten on both sides). **Put your name on it and turn it in with your exam**.

Please turn off and put away all other materials (including phones, earbuds and watches). **Do all your work on the exam itself.** There are extra blank pages at the end.

When you are done, turn in your exam and sheet of notes, and show your ID to a staff member.

Good luck!

Part I. Multiple-choice (3 pts. each)

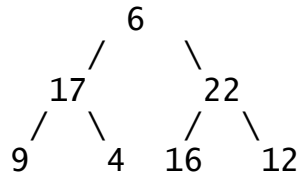
Darken the bubble for each answer in the grid on the cover sheet. ***Make sure to completely fill in the appropriate bubble, and to erase any stray marks.***

1. Suppose that items A, B, C, D and E are pushed, in that order, onto an initially empty stack S. S is then popped four times; as each item is popped off, it is inserted into an initially empty queue.

   If two items are then removed from the queue, what is the *next* item that will be removed from the queue?

   A. item A
   B. item B
   C. item C
   D. item D
   E. item E

2. If the binary tree below is printed by a preorder traversal, what will the result be?



   A. 9 4 17 16 12 11 6
   B. 9 17 6 4 16 22 12
   C. 6 9 17 4 16 22 12
   D. 6 17 22 9 4 16 12
   E. 6 17 9 4 22 16 12

3. A graph implementation that uses a two-dimensional array to represent the edges would be most reasonable for which of the following cases?

   A. 1000 nodes, 1200 edges
   B. 100 nodes, 4000 edges
   C. 1000 nodes, 10000 edges
   D. 10 nodes, 20 edges
   E. none of these, since a graph can only be represented by a linked structure.

4. A binary tree is constructed of nodes that are instances of the following class:

```
public class Node {
    public int val;
    public Node left;
    public Node right;
}
```

Consider the following method:

```
public static Node mystery(Node root) {
    if (root.right == null)
        return root;
    else
        return mystery(root.right);
}
```

You consult three supposedly tech-savvy consultants, and you get the following three opinions about what the method does when passed a reference to the root node of a binary tree:

I. It returns the last node visited by an inorder traversal
II. It returns the last node visited by a postorder traversal
III. It returns the last node visited by a level-order traversal

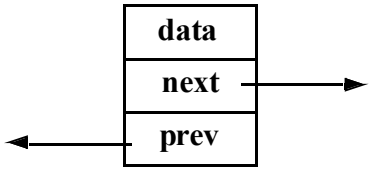
Which of these opinions is correct regardless of the contents of the tree?

A. I only
B. II only
C. III only
D. I and III
E. II and III

5. With a poorly chosen hash function, it is possible to have a situation in which the search time in a hash table of N items goes to

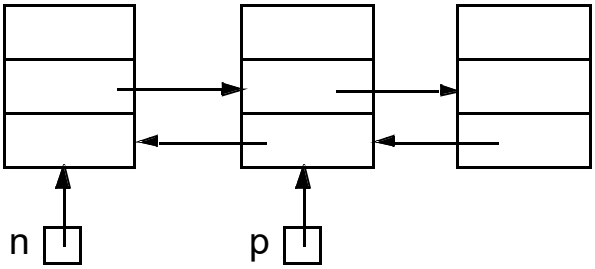
A. $O(N)$
B. $O(N!)$
C. $O(\log N)$
D. $O(N^2)$
E. $O(1)$

6. Nodes for a doubly linked list are defined to have the following structure:



The `next` instance variable stores a reference to the next node in the list, and the `prev` instance variable refers to the previous node in the list.

Below is a list of three of these nodes, along with two reference variables, `n` and `p`, that refer to specific nodes in the list.



Which of the following expressions does *not* refer to the third node in the list?

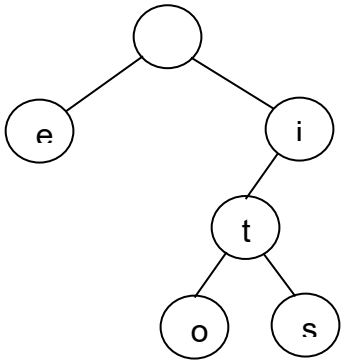
A. p.next
B. n.next.next
C. p.prev.next
D. p.next.prev.next
E. n.next.next.prev.next

7. A police department wants to maintain a database of up to 1800 license-plate numbers of people who receive frequent tickets so that it can be determined very quickly whether or not a given license plate is in the database. Speed of response is very important; efficient use of memory is also important, but not as important as speed of response. Which of the following data structures would be most appropriate for this task?

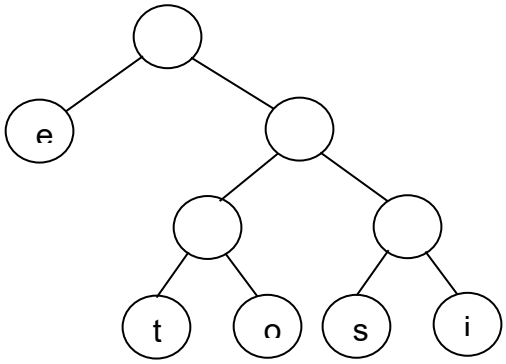
A. a sorted linked list
B. a sorted array with 1800 entries
C. a hash table using open addressing with 1800 entries
D. a hash table using open addressing with 3600 entries
E. a hash table using open addressing with 10000 entries

8. A Huffman tree is constructed for a text document containing 5 characters. The character ‘e’ appears most frequently, and the character ‘i’ has the next highest frequency. Which of the following could be the Huffman tree for this document?

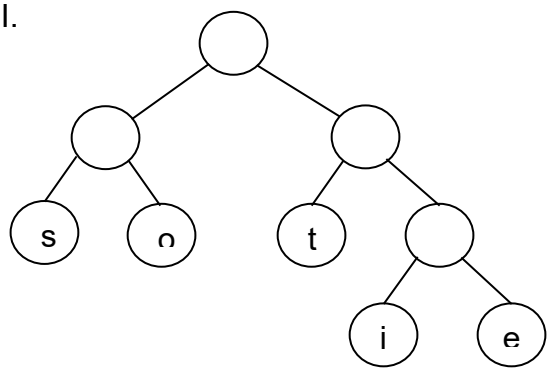
I.



II.



III.

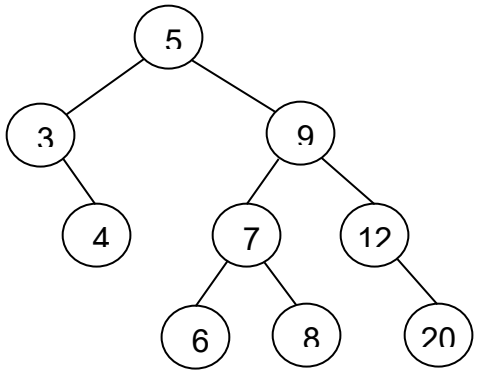


A. I only
B. II only
C. III only
D. either II or III
E. none of these

9. An array of 7 integers is being sorted by the heapsort algorithm. After the initial phase of the algorithm (constructing the heap), which of the following is a possible ordering for the array?

A. 85 78 45 51 53 47 49
B. 85 49 78 45 47 51 53
C. 85 78 49 45 47 51 53
D. 45 85 78 53 51 49 47
E. 85 51 78 53 49 47 45

10. The binary search tree shown below was constructed by inserting a sequence of items into an empty tree.



Which of the following input sequences will *not* produce this binary search tree?

A. 5 3 4 9 12 7 8 6 20
B. 5 9 3 7 6 8 4 12 20
C. 5 9 7 8 6 12 20 3 4
D. 5 9 7 3 8 12 6 4 20
E. 5 9 3 6 7 8 4 12 20

PART II: Answer 3 of the 4 problems from this part. (12 pts. each; 36 pts. total)
Important: Fill out the box accompanying each problem to indicate if it should be graded.
If you leave the boxes blank, we will *not* count your best 3 problems.

II-1. Trees *(15 points total)*

_____ GRADE _____ DON'T GRADE

a. *7 points*
Suppose the keys on the middle row of a standard keyboard (ASDFGHJKL) are inserted in succession into an initially empty binary search tree. Draw the tree after this sequence of insertions has been made.

b. *8 points*
Suppose the keys on the middle row of a standard keyboard (ASDFGHJKL) are inserted in succession into an initially empty 2-3 tree. Draw diagrams to illustrate the growth of the tree, showing the tree just before and just after all splits.

II-2. Hashing *(15 pts total; 5 pts each part)*

_____ GRADE _____ DON'T GRADE

You are given an empty hash table of size 7 that uses open addressing.
The following sequence of keys is to be inserted:

15 17 8 23 3 5

Insert these keys using each of the following approaches. If overflow occurs, say so, and indicate the element that causes the overflow.

a. h(x) = x % 7; linear probing

0	
1	
2	
3	
4	
5	
6	

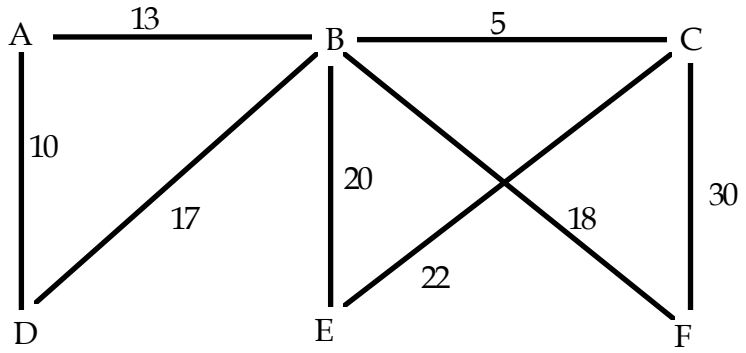
b. h(x) = x % 7; quadratic probing

0	
1	
2	
3	
4	
5	
6	

c. h(x) = x % 7; double hashing with h2(x) = x / 7 + 1 (using integer division)

0	
1	
2	
3	
4	
5	
6	

II-3. Graph Algorithms I *(15 points total)* _____ GRADE _____ DON'T GRADE

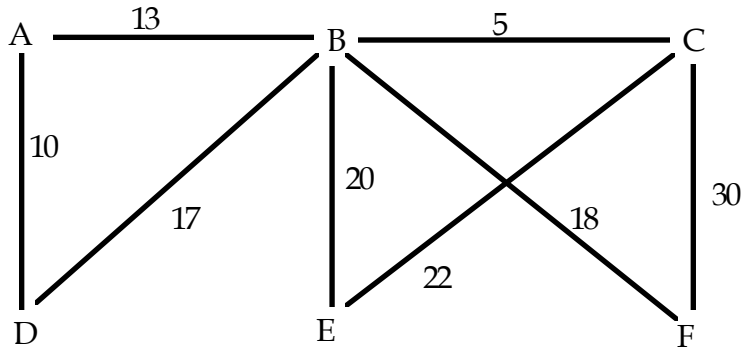


a. *4 points*
Perform a depth-first traversal of the graph shown above, starting with vertex C. Select the smallest edge first when appropriate. In the space below, list the vertices in the order in which they are visited.

b. *4 points*
Perform a breadth-first traversal of the graph shown above, starting with vertex C. Select the smallest edge first when appropriate. In the space below, list the vertices in the order in which they are visited.

c. *7 points*
Suppose you are using Dijkstra's algorithm to find the shortest path from vertex D to vertex E. List, in the order in which they become known, all vertices to which a shortest path is determined in the process of solving this problem, and the length of the shortest path to each of these vertices.

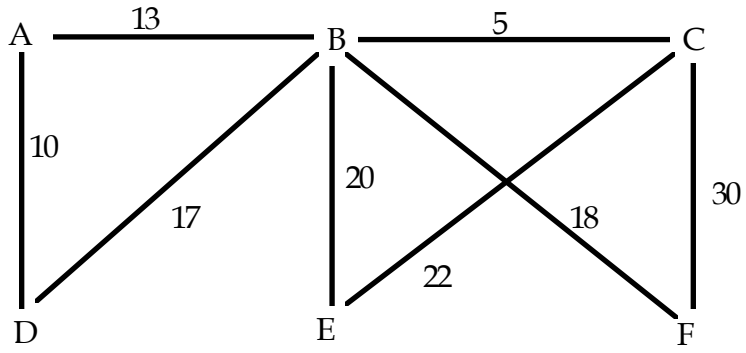
II-4. Graph Algorithms II *(15 points total)* _____ GRADE _____ DON'T GRADE



a. *6 points*
Suppose that Prim's algorithm has been executed, starting from node F, up to the point at which there are four edges selected for inclusion in the minimal spanning tree. List these four edges in the order that they are selected for inclusion, using notation similar to (A, B) to specify an edge.

b. *6 points*
Suppose that the MST2 algorithm from Problem Set 5 has been executed, up to the point where there are four edges selected for inclusion in the minimal spanning tree. List these four edges in the order that they are selected for inclusion.

c. *3 points*
Show an example of a spanning tree that is *not* minimal by darkening the appropriate edges on the diagram below:



PART III (25 pts total): Complete all of the following problems.

This part of the exam deals with binary trees that are constructed of nodes that are instances of the following class:

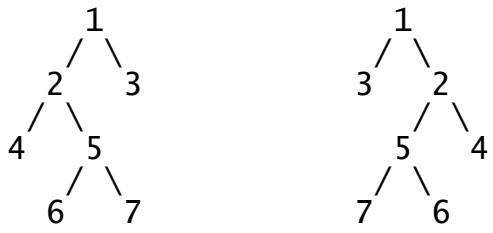
```
public class Node {
    public int key;
    public Object data;
    public Node left;
    public Node right;
}
```

a. *10 points*
The following Java method uses recursion to search for a key in the binary search tree whose root node is referred to by the parameter `root`. If it finds the key, it returns a reference to the corresponding data item. If it doesn't find it, it returns `null`.

```
public static Object search(Node root, int key) {
    if (root == null) {
        return null;
    } else if (key == root.key) {
        return root.data;
    } else if (key < root.key) {
        return searchTree(root.left, key);
    } else {
        return searchTree(root.right, key);
    }
}
```

In the space below, rewrite the `search()` method so that it uses iteration instead of recursion:

b. *10 points*
Write a Java method named `mirror()` that takes a reference to the root node of a binary tree and creates a new tree (with its own nodes) that is the mirror image of the original tree. For example: if `root` is a reference to the root of the tree on the left below, then the return value of `mirror(root)` would be a reference to the root of the tree on the right below.
Hint: This method is much easier to write if you use recursion.




```
public static Node mirror(Node root) {









}
```

c. *5 points*
What is the running time of your implementation of the `mirror()` method? Use big-O notation, and explain your answer briefly.

THE PRACTICE EXAM ENDS HERE.

Supplemental Practice Problems

1. Suppose that you need to maintain a collection of data whose contents are fixed—i.e., you need to search for and retrieve existing items, but never need to add or delete items. Although the collection of data may be quite large, you may assume that it can fit in the computer's memory. Which of the following data structures is the most efficient one to use for this task?

   A. a sorted array
   B. a linked list
   C. a binary search tree
   D. a queue
   E. all of the above perform the same in this case

2. Given the array of integers `arr` shown below

5	19	6	12	48	6	23	39	2	18

   what is the output of the following statements?

```
int[] a2 = arr;
a2[3] = arr[2];
arr[2] = arr[3];
arr[3] = a2[3];
System.out.println(arr[2] + " " + arr[3]);
```

   A. 6 12
   B. 12 6
   C. 6 6
   D. 12 12
   E. none of the above

3. If a binary search tree is not allowed to have duplicates, there is more than one way to delete a node in the tree when that node has two children. One way involves choosing a replacement node from the left subtree. If this is done, which node are we looking for?

   A. the largest node in the subtree
   B. the smallest node in the subtree
   C. the root of the left subtree
   D. the next-to-smallest node in the subtree
   E. it doesn't matter – any node in the left subtree will do

4. Which of the following data structures is most appropriate for situations in which you need to efficiently manage (key, value) pairs that are stored on disk?

   A. an array
   B. a linked list
   C. a binary search tree
   D. a 2-3 tree
   E. a B-tree

5. Consider the following recursive method:

```
public static int recurse(int a, int b) {
    if (a % b == 2) {
        return a;
    } else {
        return recurse(a + b, a - b);
    }
}
```

   What is returned by the call `recurse(7, 2)`?

   A. 5
   B. 7
   C. 9
   D. 14
   E. 18

6. Which of the following sorting algorithms does <u>not</u> require $O(n^2)$ steps in the <u>worst</u> case?

   A. insertion sort
   B. selection sort
   C. heap sort
   D. bubble sort
   E. quicksort