5CCGD003W Algorithms: In-class test

Question 1

Suppose you have the following runtime data for an algorithm. What complexity class do they indicate?

Input size	Seconds
1000	7
2000	15
4000	31
8000	63
16000	125

Example answer

The algorithm is linear. (2 points)
The input size doubles between inputs, (2 points)
and the runtimes grow by a factor of approximately 2. (2 points)
Since $2 = 2^1$ (i.e. $\log_2(2) = 1$) (2 points)
the complexity class is $O(n^1)$. (2 points)

Question 2

Consider the following code fragment. Based on its structure, what is its complexity class?

```
int s = 0;
for(int i = 0; i < n; i++)
    for(int j = 0; j < 6; j++)
        m += j;
```

Example answer

The algorithm is linear. (2 points)
The variable i runs up to n (2 points)
and j runs up to 6 which is independent of n. (2 points)
Since only one of the nested has an upper bound which grows as n, (2 points)
the overall complexity is $O(n)$. (2 points)

Question 3

Suppose we use Binary Search to find the value 3 on the following array. Which array elements get checked, in which order, and why?

index	0	1	2	3	4	5	6
value	1	2	3	5	8	13	21

Example answer

The elements that get checked are `a[3], a[1], a[2]`. (2 points)
a[3] is the middle element, and is bigger than 3; (2 points)
so we restrict to the range a[0],. . . ,a[2]) (2 points)
a[1] is the middle of the remaining range, and is smaller than 3; (2 points)
after restricting again, a[2] is the middle of the remaining range, and contains 3. (2 points)

Question 4

Suppose we run Bubble Sort to sort the following array in increasing order. What does the array look like after the first 3 iterations, and why?

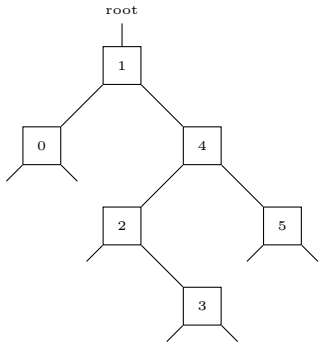
index	0	1	2	3	4	5	6
value	13	2	21	3	1	8	5

Example answer

In each iteration the we go through unsorted section (2 points)
swapping each element with the next one if they are out of order. (2 points)
In the first iteration, this results in (2, 13, 3, 1, 8, 5, 21) (2 points)
In the second iteration, this results in (2, 3, 1, 8, 5, 13, 21) (2 points)
In the third iteration, this results in (2, 1, 3, 5, 8, 13, 21) (2 points)

Question 5

Suppose we remove the value 4 from the following binary search tree. What does the resulting tree look like?



To enter your answer, write the contents of the tree one level at a time, using "*" to indicate missing nodes. For example, the above tree would be written

```
1
0  4
*  *  2  5
*  3  *  *
```

Example answer

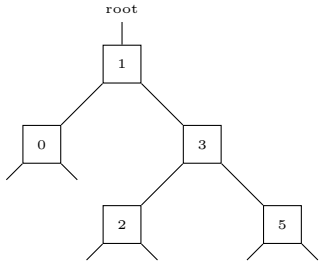
There are two possible results, depending on which value you replace the 4 with:

```
1
0  3
*  *  2  5
```

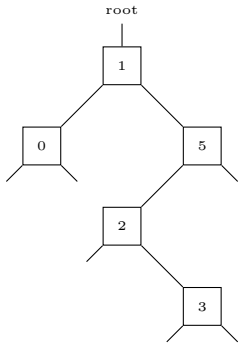
and

```
1
0  5
*  *  2  *
*  3
```

(representing the below trees; note that only the textual representation above is required)

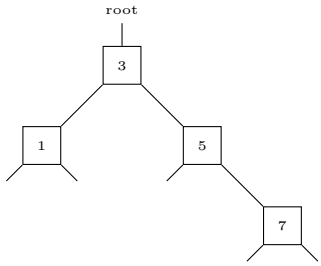


and



Question 6

Suppose we insert the value 6 in the following AVL tree, and re-balance it. What does the resulting tree look like?



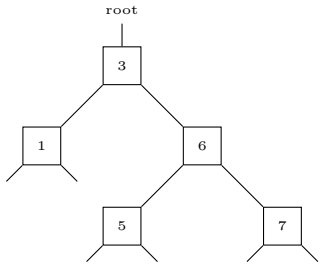
To enter your answer, write the contents of the tree one level at a time, using "*" to indicate missing nodes. For example, the above tree would be written

```
3
1   5
*   *   *   7
```

Example answer

```
3
1   6
*   *   5   7
```

(representing the below tree; note that only the textual representation above is required)



Question 7

Consider the following array representation of a Min-Heap. How does this change after inserting the value 0?

index	0	1	2	3	4
value	1	2	4	5	3

Example answer

index	0	1	2	3	4	5
value	0	2	1	5	3	4

Question 8

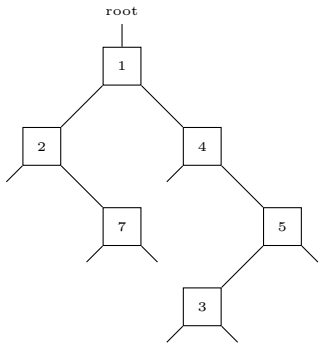
Consider the undirected graph given by
$V = \{a, b, c, d, e\}$
$E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{c, d\}, \{d, e\}\}$.
Write the adjacency matrix of this graph.

Example answer

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Question 9

Suppose we use an in-order traversal to output the following tree. What does this traversal do? What is the resulting output?



Example answer

A pre-order traversal outputs the graph recursively (2 points)
by first outputting the left subtree, (2 points)
then printing the root, (2 points)
then outputting the right subtree. (2 points)
The resulting output is 2 7 1 4 3 5. (2 points)

Question 10

For this question, consider the following problem:
Input: An array of positive integers Output: An integer that is **not** in the array
What would a brute force algorithm for this problem do? What is its complexity in terms of the size n of the array?

Example answer

A brute force algorithm first checks if there is a 1 in the array. If there is a 1, it then checks if there is a 2, and so on. (2 points)
It checks each possible value by comparing it to all values in the array. (2 points)
For each value it checks, it makes up to n comparisons. (2 points)
It will check up to $n + 1$ values before it find one that is not in the array (2 points)
for a total of $n(n + 1)$ comparisons, which is in $O(n^2)$. (2 points)