## 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 |

## 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;
```

## 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 |

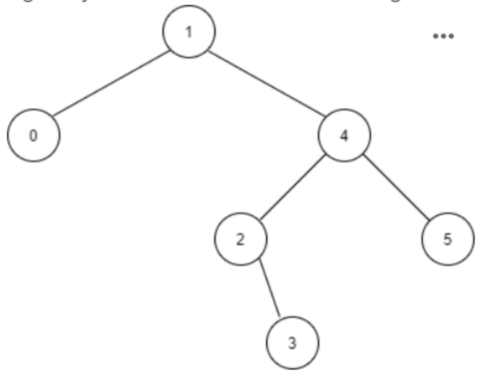
## Question 4

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 |

## Question 5

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

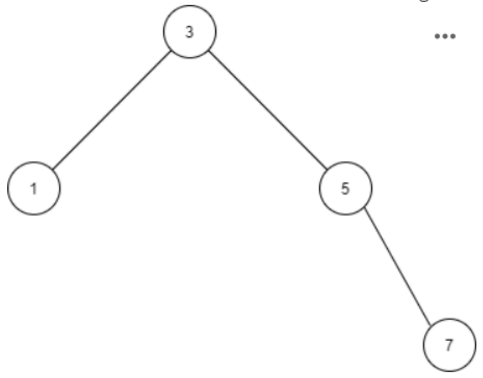


To enter your answer, write the contents of the tree one level at a time, using "*" to indicate missing nodes. For example, the above tree would be written

1
0 4
* * 2 5
* 3 * *

## 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

## Question 7

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

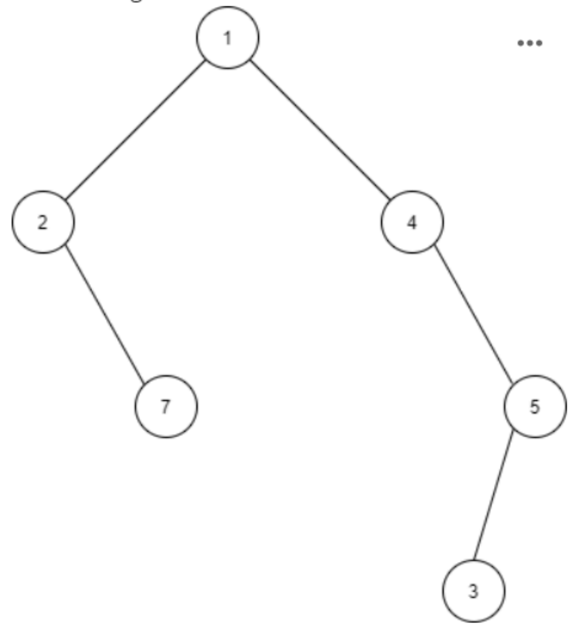
| index | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| value | 1 | 2 | 4 | 5 | 3 |

## 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.

## Question 9

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



## Question 10

For this question, consider the following problem:

Input: An array of positive integers
Output: A positive integer that is **not** in the array

What would a brute force algorithm for this problem do? What is the complexity of this algorithm in terms of the size n of the array?