Student Number (Required) ______________________

Name (Optional) ________________________________

This is a closed book test. You may not refer to any resources.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked, but will not be re-marked under any circumstances.

The test will be marked out of 50.

<table>
<thead>
<tr>
<th>Question 1</th>
<th>/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 2</td>
<td>/10</td>
</tr>
<tr>
<td>Question 3</td>
<td>/10</td>
</tr>
<tr>
<td>Question 4</td>
<td>/20</td>
</tr>
</tbody>
</table>

| TOTAL      | /50 |
Question 1 (10 marks)

a) [4 marks] Prove that $n^3$ is not in $O(n^2)$

Hint: prove there cannot exist constants $c$ and $n_0$ such that $n^3 \leq c \cdot n^2 \quad \forall n \geq n_0$

b) [4 marks] Prove that $n^2$ is not in $\Omega(n^3)$

c) [2 marks] Prove that $n^2$ is not in $\Theta(n^3)$
Question 2 (10 marks)

Suppose we have a one-dimensional array A containing 15 distinct integers in ascending order.

a) [5 marks] What is the maximum number of elements we will look at when performing Binary Search on A for an arbitrary target value x?

b) [5 marks] If x is not present in A, what is the minimum number of elements we will look at when performing Binary Search on A for x?
Question 3 (10 marks)

Let S be a stack containing positive integer values. Write an algorithm that will return the smallest value in S. S should be unchanged after your algorithm finishes. Your algorithm may use only the defined functions

```python
    S.push(x)
    S.pop()
    S.is_Empty()
```

to manipulate S.

If S is initially empty, your algorithm should return -1

Hint: your algorithm is allowed to create and use another stack.
Question 4 (20 marks)

Suppose we have a Binary Search Tree containing a set of n integers, some of which may be duplicates.

Write an algorithm called count that takes two parameters:
- t, which is a tree
- x, which is a target integer

and returns the number of vertices of the tree that contain values \( \leq x \). Your algorithm should search only as much of the tree as it needs to.

For example, if the tree t is

```
    7
   / \   \   
  4   10  
 / \   /   /
2  7 8 15
```

then \( \text{count}(t, 9) \) should return the value 5 because t contains 5 vertices with values \( \leq 9 \)

You may use iteration or recursion.

Please write your answer on the next page.
Page for answering Question 4