
DSC 40B - Quiz 04

Name:

Student ID:

About the quizzes:

- Quizzes in DSC 40B are *optional* and graded pass/fail.
- A score of 70% or higher earns a “pass” and 1.5 credits toward your final grade.
- If you don’t pass, no credits are earned, but it doesn’t hurt your grade.
- You have 30 minutes to complete the quiz.
- Unfortunately, we can’t answer clarifying questions during the quiz. If you think a question has a bug or is unclear, please let us know in a private post on Campuswire after the quiz, and we’ll take it into account when grading.

Problem 1.

True or False: `quickselect` requires the input array to be sorted in order to function correctly.

- ☐ True
- ☐ False

Problem 2.

Recall the `partition` operation from `quickselect`. Which one of the following arrays could be the result of partitioning an array around some pivot?

- ☐ [8, 3, 5, 12, 10, 1]
- ☐ [4, 2, 7, 6, 9, 8]
- ☐ [3, 1, 2, 5, 9, 7]
- ☐ [9, 7, 5, 3, 2, 1]

Problem 3.

Consider the function `foo` below.

```
def is_even(x):  
    """Returns True if x is even, False otherwise."""  
    return x % 2 == 0  
  
def foo(numbers):  
    barrier_ix = 0  
    for i in range(len(numbers)):  
        if is_even(numbers[i]):  
            numbers[barrier_ix], numbers[i] = (  
                numbers[i], numbers[barrier_ix]  
            ) # swap  
            barrier_ix += 1
```

Which one of the following is a valid loop invariant for the `for`-loop? That is, which one is true after every iteration, no matter what list `numbers` is?

- ☐ All numbers in `numbers[:barrier_ix]` are even.
- ☐ All numbers in `numbers[:barrier_ix]` are odd.
- ☐ All numbers in `numbers` are even.
- ☐ `numbers` is in sorted order.

Problem 4.

In this problem, n is the number of elements in the input array.

a) What is the **expected** time complexity of `quicksort`?

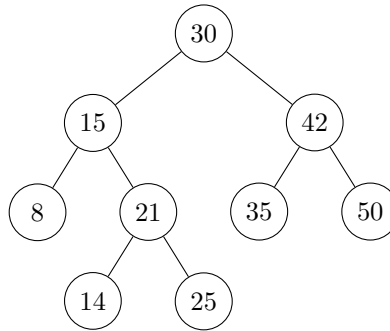
- ☐ $\Theta(\log n)$
- ☐ $\Theta(n)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n^2)$

b) What is the **worst case** time complexity of `quicksort`?

- ☐ $\Theta(\log n)$
- ☐ $\Theta(n)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n^2)$

Problem 5.

True or False: the tree below is a binary search tree.



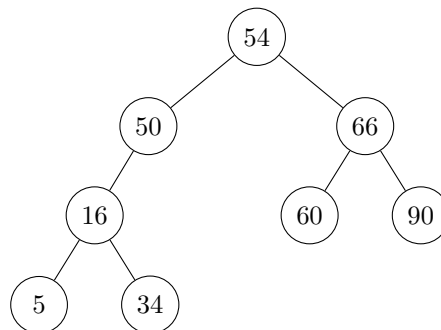
- ☐ True
- ☐ False

Problem 6.

Recall the `query` method from lecture. Below, we have added a line that prints `"Hello"`:

```
def query(self, target):
    current_node = self.root
    while current_node is not None:
        print("Hello")
        if current_node.key == target:
            return current_node
        elif current_node.key < target:
            current_node = current_node.right
        else:
            current_node = current_node.left
    return None
```

Consider the following binary search tree:

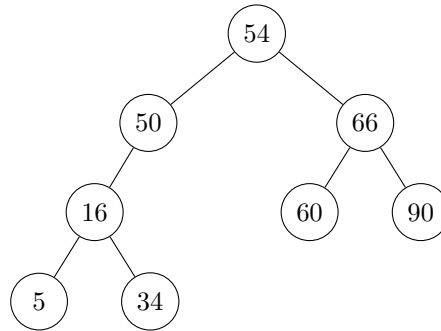


Suppose we do a query for 3 in this tree. How many times is `"Hello"` printed?

- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Problem 7.

Consider the same binary search tree as in the previous problem:



Where should we insert 49?

- ☐ the left child of 50
- ☐ the right child of 16
- ☐ the left child of 60
- ☐ the right child of 34
- ☐ the left child of 34

Problem 8.

Suppose `bst` is an empty binary search tree, and that the code below inserts n keys into it using the `insert` method from lecture.

```
for i in range(n):  
    bst.insert(i)
```

What is the time complexity of this code, as a function of n ?

- ☐ $\Theta(n)$
- ☐ $\Theta(n \log n)$
- ☐ $\Theta(n^2)$
- ☐ $\Theta(n^2 \log n)$

Problem 9.

True or False: in **any** binary search tree with n nodes, the `query` method from lecture takes $\Theta(\log n)$ time in the worst case.

- ☐ True
- ☐ False

Problem 10.

Suppose `quickselect` is used to find the **minimum** of an array of n distinct numbers (that is, $k = 1$). Which pivot would be the **worst** choice for the first call, in the sense that it leaves the largest subproblem for the next recursive call?

- ☐ The minimum.
- ☐ The second smallest number.
- ☐ The median.
- ☐ The maximum.