Quiz 4

Due Apr 27 at 1:59am **Points** 24 **Questions** 6
Available until Apr 30 at 1:59am **Time Limit** 15 Minutes **Allowed Attempts** 2

Instructions

Searching, sorting, algorithm analysis

The questions are all multiple choice or true/false. You may take the quiz twice, with 15 minutes for each attempt. Questions may differ between attempts. The highest score of the two attempts will be considered. You can only view the results once, immediately after finishing an attempt.

This is an open book test, however, you may not share the questions/answers of your quiz with others.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	7 minutes	16 out of 24

ⓘ Correct answers are hidden.

Score for this attempt: **16** out of 24
Submitted Apr 26 at 7:26pm
This attempt took 7 minutes.

Question 1 — 4 / 4 pts

According to the "Searching, sorting, algorithm analysis" lesson, the worst case for a linear search is when the target is not in the list. Why is this?

- [x] Because we have to check every item in the list, instead of being able to stop sooner when we find the item.
- [ ] Because then the search returns -1 and we have to figure out what to do with that value.
- [ ] Because the list is a waste of space if it doesn't contain the item we are looking for.
- [ ] Because it's a waste of time to look for something that isn't there.

Question 2

4 / 4 pts

Say we have a function that performs $\frac{3}{4}n^3 + \frac{1}{4}n - 1$ operations. How would we represent the time complexity of the function?

- [ ] O(n-1)
- [ ] O(2n^3)
- [ ] O(n^3 + n)
- [x] O(n^3)

Question 3

4 / 4 pts

Binary search is equally effective on a **sorted** or an **unsorted** list.

- [ ] True

- False

Question 4

4 / 4 pts

If the target item is not in a (non-empty) list, then linear search and binary search are equally time-efficient.

- True
- False

Incorrect

Question 5

0 / 4 pts

Time complexity tells us...

- How much time it will take to understand an algorithm.
- How much time it will take to run an algorithm.
- How quickly the amount of time an algorithm requires to solve a problem increases as the problem size increases.
- How many bytes of memory an algorithm requires to solve a problem.

Incorrect

Question 6

0 / 4 pts

What is the time complexity of the following function?

```
def find_first_odd(num_list):
    '''
    returns the first item in num_list that is an odd number
    if no odd numbers found, returns None
    num_list is a list of integers
    '''
    for num in num_list:
        if num%2 == 1:
            return num
    print("No odd numbers found.")
    return None
```

- O(n^2)
- O(log n)
- O(n/2)
- O(n)

Quiz Score: **16** out of 24