Learning Quiz 30: Searching

Due Nov 27 at 1pm **Points** 5 **Questions** 5 **Available** until Dec 4 at 11:59pm **Time Limit** None
Allowed Attempts Unlimited

Instructions

Prior to completing this quiz, be sure to read:

- Section 10.4: Searching (p. 354-360)

Please also go over Practice Problems 10.9-10.10 in the textbook (solutions at the end of the chapter) before attempting this quiz.

This quiz was created for learning purposes. You may attempt this quiz as many times as you would like. The highest score *prior to the deadline* will count towards the final course grade. No late submissions will be accepted.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	2 minutes	5 out of 5
LATEST	Attempt 2	2 minutes	5 out of 5
	Attempt 1	3 minutes	2 out of 5

Score for this attempt: **5** out of 5
Submitted Nov 26 at 11:58am
This attempt took 2 minutes.

Question 1

1 / 1 pts

Recall that ***in*** is a reserved word in Python. For lists, the in operator will check whether an item is in a list or not. Unless the list has any order, the most efficient way to implement a function like the in operator is to search the elements of a list one-by-one until the desired item is encountered.

Searching the elements of a list one-by-one could occur forward, from element 0 through the last element. It could also occur backward from the last element to element 0. Either way, this is called linear search.

Suppose we have the list [13, 35, 9, 23, 76, 25, 14, 7, 32]. A linear search forward for the number 20 would compare (in order):
* Is 13 equal to 20?
* Is 35 equal to 20?
* Is 9 equal to 20?

* ... ...
* Finally, is 32 equal to 20? No, so 20 is not in this list.

A linear search backward would start by comparing:
* Is 32 equal to 20?
* Is 7 equal to 20?
* Is 14 equal to 20?

What is the last comparison that would be made if searching for 20 backward in the list [13, 35, 9, 23, 76, 25, 14, 7, 32]?
* Is _____ equal to 20?

- 25
- 20
- 35
- 23
- Correct! 13
- 32

Question 2

1 / 1 pts

However, if our list is sorted, [7, 9, 13, 14, 23, 25, 32, 35, 76], we can actually do a better job searching for our desired value.

Suppose we are trying to look for the value 20. Assuming that we might not know anything about our given list, we could start by searching in the middle.
* 23 is in the middle, and it is larger than 20. Search the left half [7, 9, 13, 14]
* 13 (or 9) is in the middle, and it is smaller than 20. Search the right half [14]
* 14 is in the middle, and it is smaller than 20. Search the right half [] -- nothing to search
We can now conclude that 20 is not in the list.

By cutting our list into approximate halves, we reduce the number of comparisons we need to make to confirm whether 20 exists in our list or not. This process is called **binary search**.

The list is cut into approximate halves because, in the case of even-numbered lists like [7, 9, 13, 14] the middle is actually between 9 and 13. It is up to you whether you want to compare the left or right number.

How many comparisons would linear search have to make in searching [7, 9, 13, 14, 23, 25, 32, 35, 76] before concluding that 20 is not in the list?

Correct!

9

Correct Answers 9 (with margin: 0)

Question 3

1 / 1 pts

How many comparisons would binary search have to make in searching [7, 9, 13, 14, 23, 25, 32, 35, 59, 62, 76] before concluding that 34 is not in the list?

Correct!

4

Correct Answers 4 (with margin: 0)

Comparison I: 25 to 34
Comparison II: 59 to 34
Comparison III: 32 or 35 to 34
Comparison IV: 35 or 32 to 34

Question 4

1 / 1 pts

If a value is found, the search algorithm can quit immediately and simply say that the desired value has been found.

For example, in the list [1, 2, 3, 4, 5, 6, 7], a search for the value 2 will only have two comparisons:
* compare 4 to 2
* compare 2 to 2 --- the value 2 has been found! No need to compare any more

How many comparisons would binary search have to make in searching **['celery', 'corn', 'daikon', 'eggplant', 'jicama', 'kale', 'leek', 'mushroom', 'okra', 'onion', 'parsnip', 'potato', 'pumpkin', 'radish', 'spinach']** before concluding that 'corn' is in the list?

Correct!

3

Correct Answers 3 (with margin: 0)

Question 5

1 / 1 pts

How many comparisons would linear search have to make in searching ['celery', 'corn', 'daikon', 'eggplant', 'jicama', 'kale', 'leek', 'mushroom', 'okra', 'onion', 'parsnip', 'potato', 'pumpkin', 'radish', 'spinach'] before concluding that 'corn' is in the list?

Correct!

2

Correct Answers

2 (with margin: 0)

In this case, linear search needs to make only two comparisons. However, binary search is more efficient because we need to consider the overall algorithm. What if the search target isn't 'corn'?

Quiz Score: **5** out of 5