

We are recovering from significant hosting issues. Much of the site is functional, but currently email delivery is not. Please bear with us as we validate site functionality.



Bookmarks

- ▶ [Welcome to the edX Platform](#)
- ▶ [Entrance Survey](#)
- ▶ [Download Python and Get Motivated!](#)
- ▶ [Week 1: Python Basics](#)
- ▶ [Week 2: Simple Programs](#)
- ▶ [Week 3: Structured Types](#)
- ▶ [Week 4: Good Programming Practices](#)
- ▶ [Midterm Exam](#)
- ▶ [Week 5: Object Oriented Programming](#)

Week 6: Algorithmic Complexity > Problem Set 6 > Problem 2

Problem 2

Bookmark this page

Problem 2-1

1 point possible (graded)

Indirection, as talked about in lecture, means you have to traverse the list more than once.

☐ True☐ False

Submit

You have used 0 of 1 attempt

Problem 2-2

1 point possible (graded)

The complexity of binary search on a sorted list of n items is $O(\log n)$.


☐ True☐ False

Submit

You have used 0 of 1 attempt

▼ **Week 6:**
Algorithmic Complexity


11. Computational Complexity

[Finger Exercises](#) 

12. Searching and Sorting Algorithms

[Finger Exercises](#) 

Problem Set 6

[Problem Set due Mar 9, 2017 15:30 PST](#) 

▶ **Week 7:**
Plotting

▶ **Exit Survey**

▶ **Sandbox**

Problem 2-3

1 point possible (graded)

The worst case time complexity for selection sort is $O(n^2)$.

☐ True

☐ False

Submit

You have used 0 of 1 attempt

Problem 2-4

1 point possible (graded)

The base case for the recursive version of merge sort from lecture is checking ONLY for the list being empty.

☐ True

☐ False

Submit

You have used 0 of 1 attempt

Problem 2

Topic: Problem Set 6 / Problem 2

Show Discussion

© All Rights Reserved



© 2012-2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX

