

A template with a preliminary implementation is provided in BeachBoard (template.zip) under *Template* of the *Content Tab*. The template provides the classes and a simple menu to interact. Modify the menu to support the required functionalities. Read the readme.txt file for the documentation.

Note Only assignments that use the template will be graded.

LAB 6: SORTING

Learning objectives: CLO 3, CLO 4

Use Python 3.8 or higher for the assignment:

1. Implement MergeSort and QuickSort seen in class.

Learning objectives: CLO 3

Test your program:

- Sort in MergeSort and QuickSort an empty array
- Sort in MergeSort and QuickSort an array of size 5 with values 4, 1, 3, 5, 2 and check that the output is 1, 2, 3, 4, 5

2. Implement the binary search algorithm seen in class.

Learning objectives: CLO 3

Test your program:

- Search in an empty array.
- Search for the elements 0, 1, 3, 5 in a sorted array 1, 2, 3, 4, 5

3. Book Store System. Using Lab 1, load the catalog "books.txt" in an instance *bookSortedCatalog* of your *ArrayList*. For each row, create an instance of the *SortableBook* that allows comparing by title and insert in *bookSortedCatalog*. In development time, use the file "booktest.txt" with few books. Once you think it is ready, use the main file "books.txt".

Learning objectives: CLO 3, CLO 4

- Sort bookCatalog using MergeSort and QuickSort. Add menu options for each.
- Search books by prefix using Binary Search: Given a prefix *prefix*, find all books in the sorted *bookCatalog* that start with *prefix* using Binary Search and display in the screen, including the index number and title.

Test your program: Searching for books by:

- Empty prefix.
- "Tears of the S"
- "World of P"

4. What is the advantage and disadvantage of Sorting and BinarySearch over BinarySearchTree and heaps data structures.

Learning objectives: CLO 4

Hint: Consider offline algorithms vs on-line algorithms

5. Bonus points (3 points). Implement MergeSort and QuickSort to sort linked-lists

Learning objectives: CLO 3, CLO 4

Hint: Consider using a list as a value.

Test your program: Searching for books by:

- Empty prefix.
- "Tears of the S"
- "World of P"

Submit all the source code (Python files (.py) in a zip file. The name of the zip file with the source code must be your first name, second name, and the data structure separated by a hyphen. For example, oscar-ponce-sorting.zip.

Submissions that do not follow the previous specification will be rejected and you will have 0 in the lab.

RUBRICS

| | Level 4 2 Pt | Level 3 1.5 Pt | Level 2 1 Pt | Level 1 0.5 Pt |
|---|--------------------------------------|---|---|---------------------------------|
| MergeSort and QuickSort implementation | It is always correct without crashes | Eventually it crashes or return incorrect results | It frequently crashes and/or return incorrect results | It is not correct or incomplete |
| Binary Search implementation | It is always correct without crashes | Eventually it crashes or return incorrect results | It frequently crashes and/or return incorrect results | It is not correct or incomplete |
| Searching books by prefix | It is always correct without crashes | Eventually it crashes or return incorrect results | It frequently crashes and/or return incorrect results | It is not correct or incomplete |
| Answer to Question 4 | N/A | N/A | Correct | Incorrect |
| MergeSort and QuickSort implementation for linked lists | It is always correct without crashes | Eventually it crashes or return incorrect results | It frequently crashes and/or return incorrect results | It is not correct or incomplete |