

PG2 –RECURSION, SORTING, SEARCHING

CONTENTS

Objectives	1
Topics Covered	1
Part 3-1	2
Project Setup	2
Part 3-2	2
Load the file.....	2
Part 3-3	2
Bubble Sort.....	2
Part 3-4	3
Binary Search.....	3
Part 3-5	4
Merge Sort	4
Part 3-6	5
The Menu	5
Programmer’s Challenge	6
Sorting Challenge	6

OBJECTIVES

Learn to load and parse CSV files. Implement various sorting and search algorithms. Utilize recursion to solve problems. Clone arrays into new lists.

You will be loading in a CSV file that contains unsorted data and you can store that information in a List (each item is a string). The user will be able to sort that data using the different sorting algorithms that we covered in the lecture: Bubble, Merge. Search a sorted list with a binary search algorithm.

Topics Covered

Cloning, bubble sort, merge sort, binary search, CSV, Split, File loading

PART 3-1

Project Setup

- Create a console application
- Add the **inputFile.csv** file to your console app.
 - Right-click the console application project
 - Select Add->Existing Item...
 - Navigate to where the file is stored on your hard drive
 - Double-click to add to your project.
 - Select the file in your Solution Explorer
 - Open the Properties tab
 - Set these properties
 - **Build Action** is **None**
 - **Copy to Output Directory** is **Copy Always**

PART 3-2

Load the file

Open and read the line from the **inputFile.csv** file. The line in the file contains a list of comic book titles separated by commas. Split the string and store each title in a **List of strings**.

PART 3-3

Bubble Sort

Implement the [Bubble sort](#) algorithm. You want to keep the original list unsorted so make sure to clone the original list each time you call the Bubble sort.

Here is the Wikipedia pseudocode:

```
procedure bubbleSort(A : list of sortable items)
  n := length(A)
  repeat
    swapped := false
```

```
for i := 1 to n - 1 inclusive do
  if A[i - 1] > A[i] then
    swap(A[i - 1], A[i])
    swapped = true
  end if
end for
n := n - 1
until not swapped
end procedure
```

Turn this into C# code. NOTE: **swap** is a method you must create. See the lectures slides for how to swap 2 items in a list.

PART 3-4

Binary Search

Implement the [Binary Search](#) algorithm (use a recursive approach).

1. Clone the original list and sort the cloned list (call Sort on the list).
2. Loop over the sorted list.
3. Call **your** binary search method to search the sorted list for each title in the sorted list.

HINT: the index returned from your binary search should match the index.

Show the search title, the index and the index returned by your binary search method.

EXAMPLE OUTPUT:

Anarky	Index: 0	Found Index: 0
Arkham Manor	Index: 1	Found Index: 1
Azrael	Index: 2	Found Index: 2
Azrael volume 2	Index: 3	Found Index: 3
Azrael: Agent of the Bat	Index: 4	Found Index: 4
Batgirl	Index: 5	Found Index: 5
Batgirl volume 3	Index: 6	Found Index: 6
Batgirl volume 4	Index: 7	Found Index: 7
Batman	Index: 8	Found Index: 8
Batman '66	Index: 9	Found Index: 9

PART 3-5

Merge Sort

Implement the [Merge sort](#) algorithm. You want to keep the original list unsorted so make sure to clone the original list each time you call the Merge sort.

Here is the Wikipedia pseudocode:

```
function merge_sort(list m) is
    // Base case. A list of zero or one elements is sorted, by definition.
    if length of m  $\leq$  1 then
        return m

    // Recursive case. First, divide the list into equal-sized sublists
    // consisting of the first half and second half of the list.
    // This assumes lists start at index 0.
    var left := empty list
    var right := empty list
    for each x with index i in m do
        if i < (length of m)/2 then
            add x to left
        else
            add x to right

    // Recursively sort both sublists.
    left := merge_sort(left)
    right := merge_sort(right)

    // Then merge the now-sorted sublists.
    return merge(left, right)
```

```
function merge(left, right) is
    var result := empty list

    while left is not empty and right is not empty do
        if first(left)  $\leq$  first(right) then
            add first(left) to result
            left := rest(left) //remove the first item
        else
            add first(right) to result
            right := rest(right) //remove the first item

    // Either left or right may have elements left; consume them.
    // (Only one of the following loops will actually be entered.)
    while left is not empty do
```

```

    append first(left) to result
    left := rest(left)
    while right is not empty do
        append first(right) to result
        right := rest(right)
    return result

```

PART 3-6

The Menu

Show a menu to the user so they can select one of the algorithms (bubble, merge and binary search) as well as Exit. (Maybe use the ReadChoice method you created in the first lab?) After calling any of the sorting methods, you should **display** the unsorted list along with the sorted list.

1. Bubble Sort
2. Merge Sort
3. Binary Search
4. Exit

EXAMPLE OUTPUT:

```

Bubble Sort
-----
Detective Comics      Anarky
Batman               Arkham Manor
World's Finest Comics Azrael
Star-Spangled Comics Azrael volume 2
The Brave and the Bold Azrael: Agent of the Bat
The Joker            Batgirl
Batman Family        Batgirl volume 3
Man-Bat              Batgirl volume 4
Batman and the Outsiders Batman
The Outsiders        Batman '66
The Adventures of the Outsiders Batman 80-Page Giant
The New 52           Batman Adventures volume 2

```

The items on the left are the unsorted values and the items on the right are the sorted.

PROGRAMMER'S CHALLENGE

As with every programmer's challenge, remember the following...

1. Do the rubric first. Make sure you have something to turn in for the assignment.
2. When attempting the challenge, don't break your other code.
3. You have other assignments so don't sacrifice them to work on the challenges.

Sorting Challenge

Add the ability to sort the list in the opposite direction. This is an alternating behavior such that one time, the sort is ascending. Then the next time that same sort method is selected, it sorts in descending order. It alternates between ascending and descending.

```
Batman volume 2
Batman The Dark Knight volume
Batman and Robin volume 2
Batwing
Birds of Prey volume 3
Catwoman volume 4
Detective Comics volume 2
Nightwing volume 3
Red Hood and the Outlaws
Batman Arkham Unhinged
Batman Incorporated volume 2
Talon
Legends of the Dark Knight volume 2
Batman Li'l Gotham
Batman '66
Harley Quinn
Grayson
Batman Superman
Batman Eternal
Gotham Academy
Gotham by Midnight
Arkham Manor
Batman Beyond
Red Hood Arsenal
Robin Son of Batman
We Are Robin
Batman and Robin Eternal
-----
Press any key to continue
```

```
Batman Incorporated
Batman Gotham Knights
Batman Gotham Adventures
Batman Family
Batman Eternal
Batman Confidential
Batman Beyond volume 4
Batman Beyond volume 2
Batman Beyond
Batman Arkham Unhinged
Batman and the Outsiders vo
Batman and the Outsiders
Batman and Robin volume 2
Batman and Robin Eternal
Batman and Robin
Batman Adventures volume 2
Batman 80-Page Giant
Batman '66
Batman
Batgirl volume 4
Batgirl volume 3
Batgirl
Azrael volume 2
Azrael Agent of the Bat
Azrael
Arkham Manor
Anarky
```