

Data Structures & Algorithms 2

Lab 1

Programming : General overview

Exercise 1

Write a program to solve the word puzzle problem (Figure 1). The word puzzle problem is to find given words in a two-dimensional array of characters. This problem has two inputs: a two-dimensional array of characters M, and a list of words that you have to find in M. These words may be horizontal, vertical, or diagonal in any direction (for a total of eight directions).

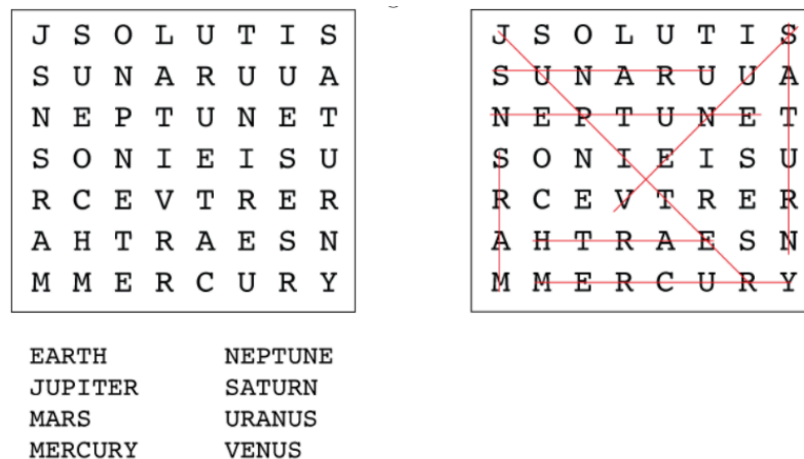


Figure 1

Exercise 2

For a given string (Example , "ABC"), write a recursive function to print to the console all possible permutations of the given string.

(→ ABC ACB BAC BCA CBA CAB)

Exercise 3

Design a class template **Collection** that stores a collection of **Objects** in an array, along with the current size of the collection. Provide the public functions *isEmpty*, *makeEmpty*, *insert*, *remove*, and *contains*. *contains(x)* returns true if an object that is equal to x is present in the collection.

(NB: Sections 1.6.2, 1.6.3 pp. 38-40 of the textbook)

Exercise 4

Design a class template ***OrderedCollection*** that stores a collection of ***Comparables*** in an array, along with the current size of the collection. Provide the public functions *isEmpty*, *makeEmpty*, *insert*, *remove*, *findMin*, and *findMax*. The functions *findMin* and *findMax* respectively return references to the smallest and largest comparable in the collection. Explain what can be done if these operations are performed on an empty collection.

(NB: Sections 1.6.2, 1.6.3 pp. 38-40 of the textbook)

Exercise 5

Write a program to solve the selection problem. The selection problem determines the k^{th} largest element from a set of N numbers. It can be solved using the following steps:

1. Initialize an array of size N with elements
2. Apply bubble sort in descending order to the elements
3. Return the K^{th} element from the array
4. Let $k = N/2$. Draw a table showing the running time of your program for various values of N .