Version: 1.0

Binary Search Visualizer

Technical Documentation.

Table of Contents

Change History

Version: 1.0
Developers: Parvindar Singh (170101044)
Sourabh Jangid (170101067)
Vakul Gupta (170101076)
Date: 22/01/2019
Description of Change: 1st edition

---

1 Introduction

Architecture and Design

The code for this software is totally written in Visual Basic in Visual Studio 2013. The software is designed in a very user friendly way. Anyone (college students) should be able to use this software to visualize binary search. The software currently has three input boxes. These are the names used for these textboxes inside the code.

Input Textboxes-

1. INPUT(textbox) – take input of number of elements.
2. SEARCH (textbox) –take input of element to be searched.
3. ARRAY (textbox) – take space separated similar elements of any datatype.

It contains three buttons.

1. Start button- starts searching the required element in the array.
2. Reset button – it resets the texts of all the textboxes and global variables declared inside the code.
3. Take input from file – takes input from file.

It has one checkbox named **AUTO**.

If it is checked, then the software shows each iteration of binary search after delay of 1 sec automatically.

If it is unchecked then we can proceed to next iteration by clicking the NEXT (After first iteration, the text of START is set to NEXT) button.

There are currently Five output Textboxes. They are disabled so that user cannot input any data in it.

1. ARRAY_2(rich textbox) - The Visualization of each iteration is shown in a separate textbox of type rich textbox.

2. LEFT (textbox)- it shows the index of the leftmost element of the new subarray after each iteration.

3. RIGHT (textbox)- it shows the index of the rightmost element of the new subarray after each iteration.

4. MID (textbox) – It shows the index of the middle element (Left + right)/2 of the new subarray after each iteration.

5. RESULT (textbox) – It is docked into a panel which is of blue color by default. If the element is found, the panel color changes to green and it prints found and shows the index of the element in the given array. If element is not found then it prints “element not found” and panel color changes to red.

2 Design Goals

The GUI design of the software is focuses on the school and college students who would use this software to understand and visualize the mechanism of binary search.
The Coding is expected to be as efficient as possible.
The design priorities for this Binary Search Visualizer application are:

- The design should minimize complexity and development effort.
- The GUI design should be Easy to understand. The first time user should know what to do in the software.

3 System Behavior

As the name suggests, It shows the binary search algorithm in a visual way.
It takes inputs from three textboxes(size, element to be searched, array of elements) .
Elements can be of mainly TWO similar datatypes – Decimals(also contains Integers) or String.
String are considered only of Upper and Lowercase English alphabets.
It takes space separated elements and converts them internally into array of size – number of elements.

Constraints checked inside the current code are –

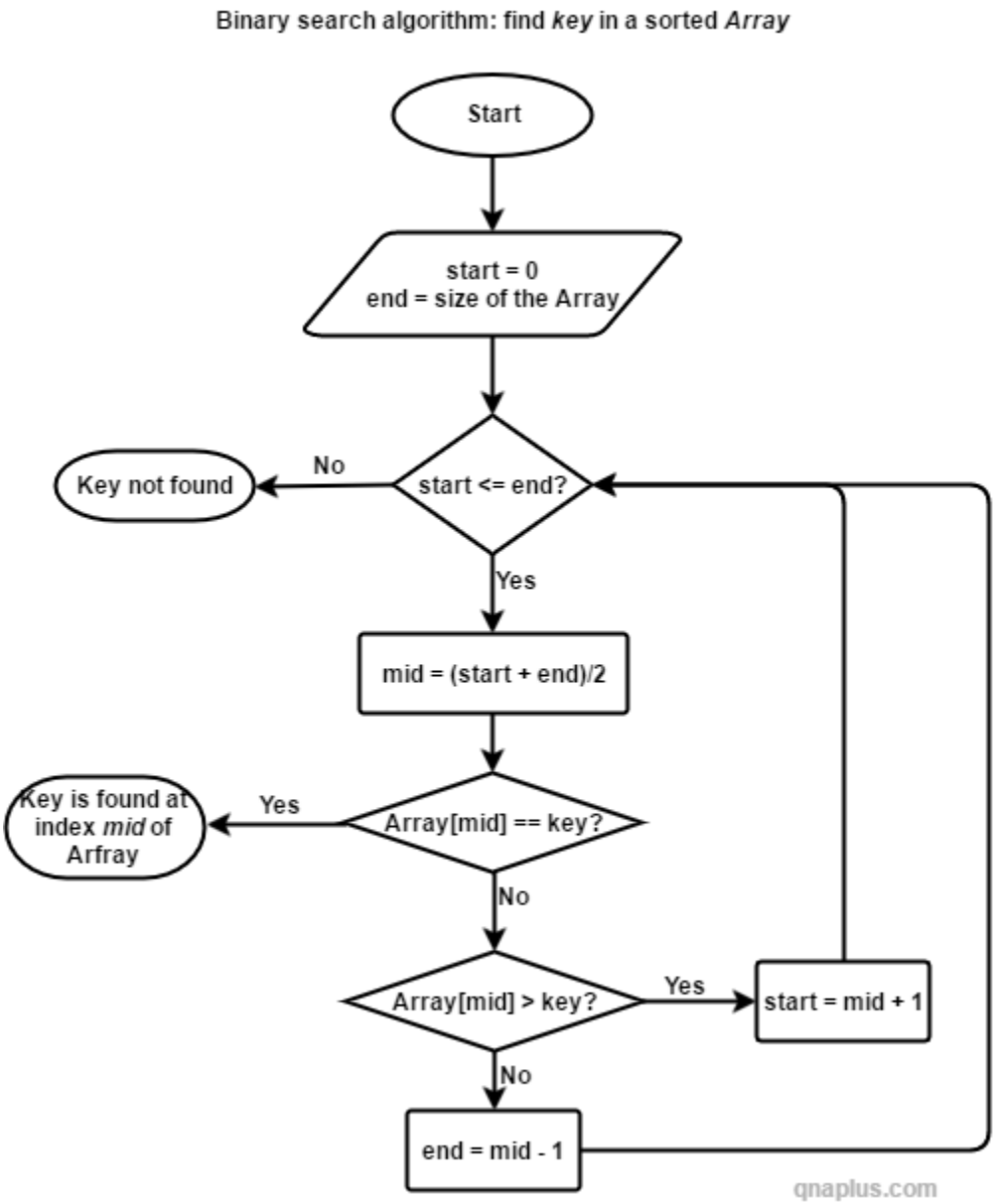
1. Size of element should be an integer.
2. Datatype mismatching .
3. Number of space seperated elements should be equal to the input size.
4. The space separated elements should be in increasing order or decreasing order.
5. Strings should contain only lowercase and uppercase English alphabets.

User can enter any number of spaces between the elements in the array textbox.

If **Auto** is Unchecked –
After clicking start , the left ,right and mid index of the array are shown in their respective textboxes. And array of elements is shown in the output textbox .
In the array_2 output textbox – Leftmost and Rightmost element is shown in Light Blue color. And mid element is shown in Orange color.
After Clicking Next(after first iteration , TEXT of Start changes to NEXT).
It Compares the middle element with required element and takes the favourable half subarray in which the element is likely to be present and color the other elements gray.
New left,right,mid are updated.
After again clicking Next.
It compares the middle of the new mid with required element and again take the favourable half subarray.
And so on….until the element is found or there is only one element is left in array.
If the element is found, the panel attached with result textbox color changes to green and it prints found and shows the index of the element in the given array. If element is not found then it prints “element not found” and panel color changes to red.

If **auto** is checked –
We only have to click on the start button once and then the process of iteration will continue with some delay.
Until element is found or only one element is left in the array.

4 Logical View



(FLOWCHART OF THE BASIC ALGORITHM CONSIDERING INPUT SORTED IN **INCREASING** ORDER)

Algorithm Used :- Binary Search algorithm

Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one. We basically ignore half of the elements just after one comparison. It works on the fact that the list is sorted either in increasing or decreasing order enabling us to overlook one half of the array after every iteration.

Running Time of the Algorithm:-

Space Complexity:- O(1)

Time Complexity:- O(logN) where N is the total number of entries

OTHER UTILITY FUNCTIONS USED:-

1. **DELAY:-**This function is used to pause the program for a few seconds after every iteration. It contains one argument i.e. the number of seconds for which the program should pause during execution. Its return type is void i.e. it does not return anything.
2. **RESET:**-This function is used to clear the contents of all the textboxes required. It also initializes all the global variables and flags used to their respective default values in order to maintain the smooth execution in future. It has zero number of arguments and it's return type is also void.
3. **INDEX:**-This function is used to calculate the index of the first character of every element which is used for highlighting with different colors to create the visualization effect .This function is called only after the input string has been processed(trimming of spaces and all that stuff) .It returns an array of indices.
4. **HIGHLIGHT**:- This function is used to color selected range (w.r.t the indexing) of the list of elements. It highlights with three different colors:-aqua, orange and grey required for different purpose. It has no arguments and it's return type is also void.
5. **FILE READING FUNCTION**:-This function opens a dialogue box allowing the user to browse through the system to select the desired file for taking input. Input format in the file is specified with the help of a Message Box which pops up on calling it.