|  |  |  |  |
| --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Student Name: Mohammad Faiz Nishat Parvej Saiyad | | | |
| Class: TY | Division: A | | Roll No:371034 |
| Semester: V | | Academic Year:2022-23 | |
| Subject Name & Code: Design and Analysis of Algorithms | | | |
| Title of Assignment: Write a program to perform binary search on an unsorted random list of at least 5000 elements. The key element should be user input. Use the Divide & Conquer method to implement this program. | | | |
| Date of Performance: | | Date of Submission: | |

**Aim:**

**Write a program to perform binary search on an unsorted random list of at least 5000 elements. The key element should be user input. Use the Divide & Conquer method to implement this program**

**Problem Statement:**

To Implement Binary Search on unsorted Array Using Divide and Conquer method

**Software Requirements:**

Text Editor: VSCode, Neovim, etc

Environment: Python 3.10

Terminal Emulator

**Background Information:**

**Divide and Conquer :-**

A divide and conquer algorithm is a strategy of solving a large problem by

breaking the problem into smaller sub-problems solving the sub-problems, and

combining them to get the desired output.

Here are the steps involved:

1. Divide: Divide the given problem into sub-problems using recursion.

2. Conquer: Solve the smaller sub-problems recursively. If the subproblem

is small enough, then solve it directly.

3. Combine: Combine the solutions of the sub-problems that are part of

the recursive process to solve the actual problem.

**Binary Search:-**

Binary Search is a searching algorithm used in a sorted array by repeatedly

dividing the search interval in half. The idea of binary search is to use the

information that the array is sorted and reduce the time complexity to O(Log

n).

Binary Search Algorithm: The basic steps to perform Binary Search are:

• Begin with the mid element of the whole array as a search key.

• If the value of the search key is equal to the item then return an index of

the search key.

• Or if the value of the search key is less than the item in the middle of the

interval, narrow the interval to the lower half.

• Otherwise, narrow it to the upper half.

• Repeatedly check from the second point until the value is found or the

interval is empty.

**Recursive Approach for Binary Search -**

binarySearch(arr, x, low, high)

if low > high

return False

else

mid = (low + high) / 2

if x == arr[mid]

return mid

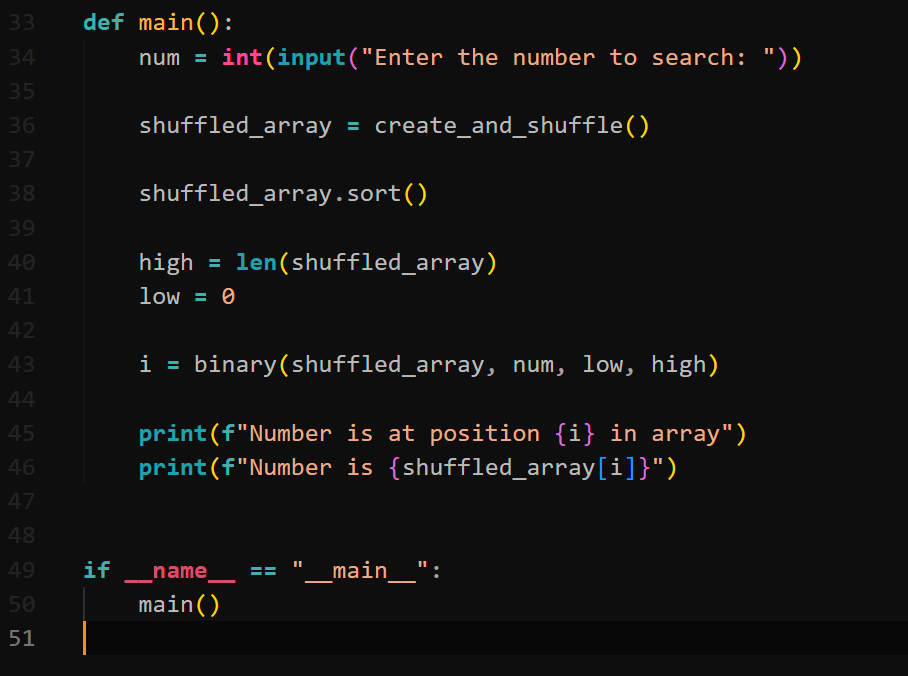
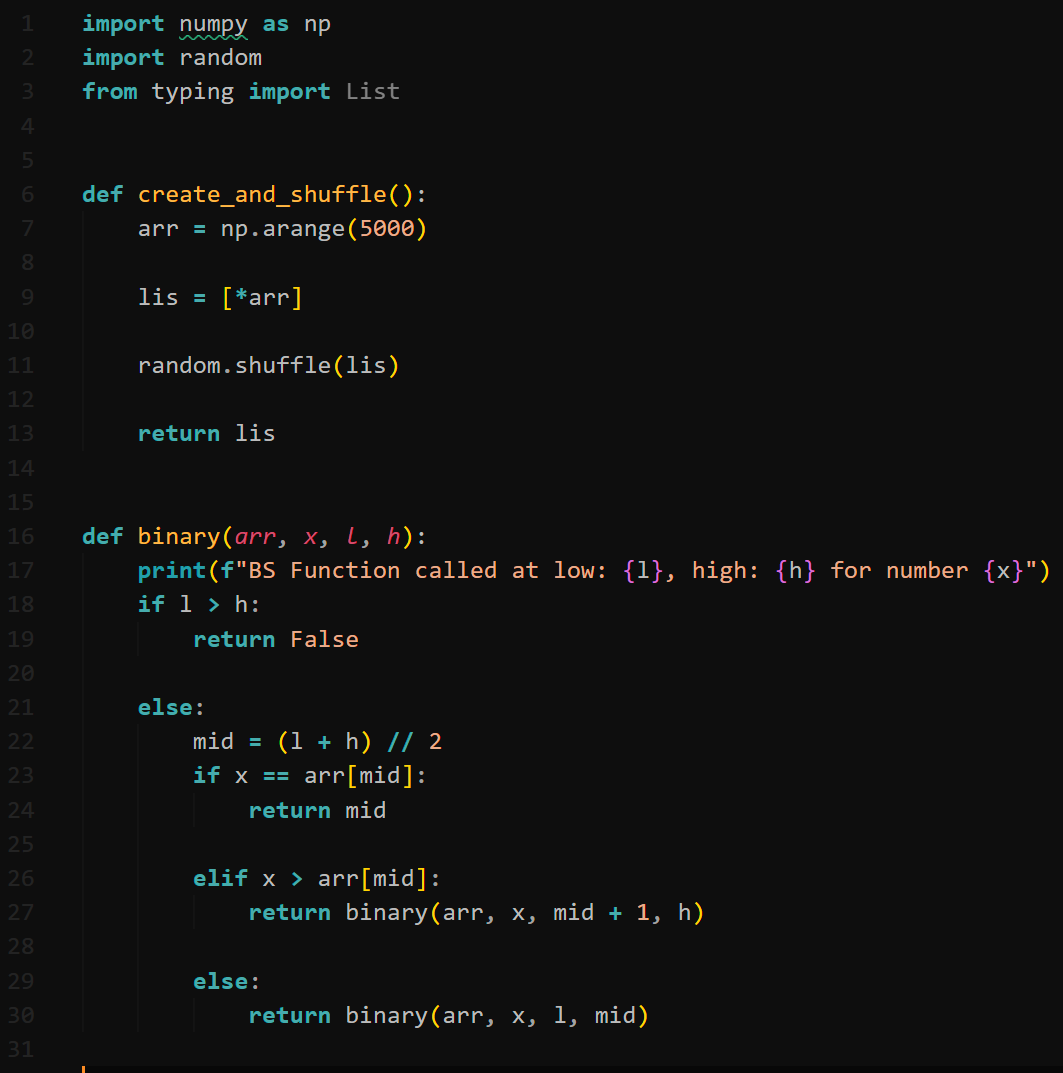
else if x > arr[mid] // x is on the right side

return binarySearch(arr, x, mid + 1, high)

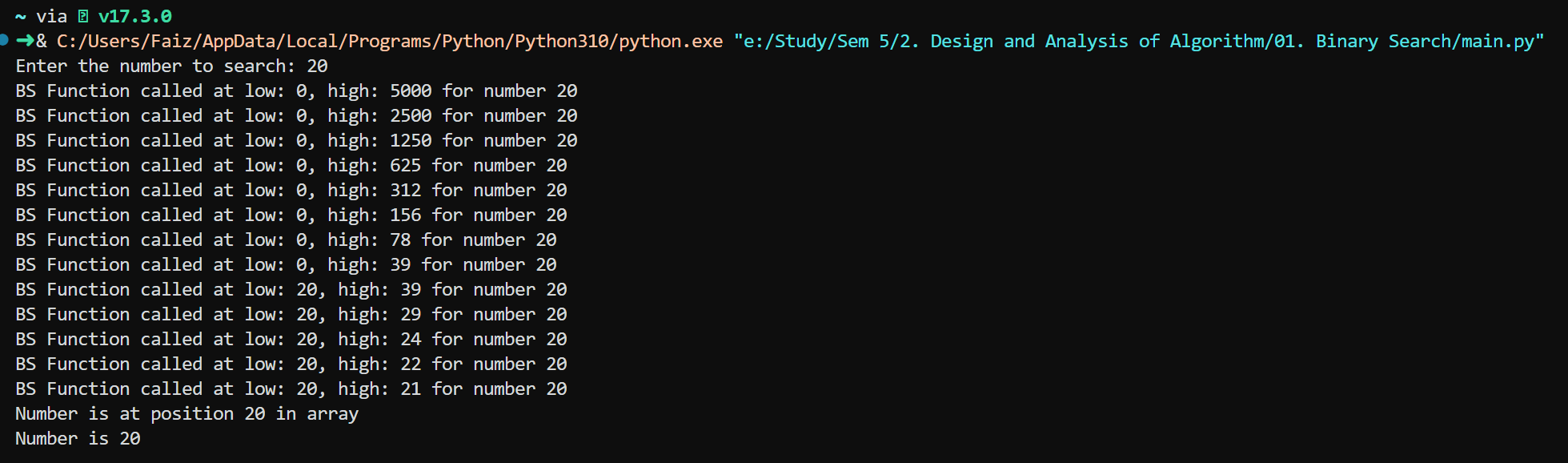
else // x is on the right side

return binarySearch(arr, x, low, mid - 1)

**Code:**



**Output:**



**Conclusion:**

Implemented Binary Search using Divide and conquer Strategy on Unsorted Array