

STACK AND QUEUE

What is the Divide and Conquer Strategy?

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

1. breaking the problem into smaller sub-problems
2. solving the sub-problems, and
3. combining them to get the desired output.

To use the divide and conquer algorithm, **recursion** is used. Learn about recursion

What is binary Search and how does it work?

Step-by-step Binary Search Algorithm: We basically ignore half of the elements just after one comparison.

1. Compare x with the middle element.
2. If x matches with the middle element, we return the mid index.
3. Else If x is greater than the mid element, then x can only lie in the right half subarray after the mid element. So we recur for the right half.
4. Else (x is smaller) recur for the left half.

1. # Iterative Binary Search Function method Python Implementation
2. # It returns index of n in given list1 if present,
3. # else returns -1
4. def binary_search(list1, n):
5. low = 0
6. high = len(list1) - 1
7. mid = 0
- 8.
9. while low <= high:
10. # for get integer result
11. mid = (high + low) // 2
- 12.
13. # Check if n is present at mid
14. if list1[mid] < n:
15. low = mid + 1
- 16.
17. # If n is greater, compare to the right of mid
18. elif list1[mid] > n:
19. high = mid - 1
- 20.

```

21.     # If n is smaller, compared to the left of mid
22.     else:
23.         return mid
24.
25.         # element was not present in the list, return -1
26.     return -1
27.
28.
29. # Initial list1
30. list1 = [12, 24, 32, 39, 45, 50, 54]
31. n = 45
32.
33. # Function call
34. result = binary_search(list1, n)
35.
36. if result != -1:
37.     print("Element is present at index", str(result))
38. else:
39.     print("Element is not present in list1")

```

Explain the distinction between a list and a tuple.

List	Tuples
Lists are mutable	Tuples are immutable.
Implication of iterations is time consuming	The implication of iteration is comparatively faster.
The list is better for programming questions such as insertion and deletion	Tuples data type is appropriate for accessing the elements.
List consumes more memory	Tuples consumes less memory as compared to list.
List have several built in methods	Tuple does not have many built-in methods.

Can you explain how Python manages memory?

- Memory management in Python involves a private heap containing all Python objects and data structures. Interpreter takes care of Python heap and that the programmer has no access to it.
- The allocation of heap space for Python objects is done by Python memory manager. The core API of Python provides some tools for the programmer to code reliable and more robust program.
- Python also has a build-in garbage collector which recycles all the unused memory. When an object is no longer referenced by the program, the heap space it occupies can be freed. The garbage collector determines objects which are no longer referenced by the sprogram frees the occupied memory and make it available to the heap space.
- The gc module defines functions to enable /disable garbage collector:

What is the difference between pickling and unpickling?

Pickling: It is a process where a Python object hierarchy is converted into a byte stream.

Unpickling: It is the inverse of Pickling process where a byte stream is converted into an object hierarchy

What are the different types of search algorithms?

1. [Linear Search](#)
2. [Binary Search](#)
3. [Jump Search](#)
4. [Interpolation Search](#)
5. [Exponential Search](#)
6. [Sublist Search \(Search a linked list in another list\)](#)
7. [Fibonacci Search](#)
8. [The Ubiquitous Binary Search](#)
9. [Recursive program to linearly search an element in a given array](#)
10. [Recursive function to do substring search](#)
11. [Unbounded Binary Search Example \(Find the point where a monotonically increasing function becomes positive first time\)](#)