

Data Structures and Algorithms Assignment 4

1. What is the divide and conquer strategy?

ANS:

Divide And Conquer

This technique can be divided into the following three parts:

Divide: This involves dividing the problem into smaller sub-problems.

Conquer: Solve sub-problems by calling recursively until solved.

Combine: Combine the sub-problems to get the final solution of the whole problem.

2. What is binary search and how does it work?

ANS:

Binary search is a fast search algorithm with run-time complexity of $O(\log n)$. This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form.

Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero

Working:

- Compare x with the middle element.
- If x matches with the middle element, we return the mid index.
- 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.
- Else (x is smaller) recur for the left half.

3. Explain the distinction between a list and a tuple.

ANS:

List

- Lists are mutable
- Implication of iterations is Time-consuming
- The list is better for performing operations, such as insertion and deletion. Tuple data type is appropriate for accessing the elements
- Lists consume more memory
- Lists have several built-in methods

Tuple

- Tuples are immutable
- The implication of iterations is comparatively Faster
- Tuple consume less memory as compared to the list
- Tuple does not have many built-in methods.

4. Can you explain how Python manages memory?

ANS:

Python uses the dynamic memory allocation which is managed by the Heap data structure.

Memory Heap holds the objects and other data structures that will be used in the program.

Python memory manager manages the allocation or de-allocation of the heap memory space through the API functions.

Garbage Collection:

Garbage collection is a process in which the interpreter frees up the memory when not in use to make it available for other objects.

Assume a case where no reference is pointing to an object in memory i.e. it is not in use so, the virtual machine has a garbage collector that automatically deletes that object from the heap memory

Reference Counting:

Reference counting works by counting the number of times an object is referenced by other objects in the system. When references to an object are removed, the reference count for an object is decremented. When the reference count becomes zero, the object is deallocated.

For example, Let's suppose there are two or more variables that have the same value, so, what Python virtual machine does is, rather than creating another object of the same value in the private heap, it actually makes the second variable point to that originally existing value in the private heap. Therefore, in the case of classes, having a number of references may occupy a large amount of space in the memory, in such a case referencing counting is highly beneficial to preserve the memory to be available for other objects

5. What is the difference between pickling and unpickling?

ANS:

Pickling: pickling is a process where a python object hierarchy is converted into a byte stream.

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

6. What are the different types of search algorithms?

ANS:

- Linear Search
- Binary Search
- Sublist Search
- Interpolation Search
- Exponential Search
- Fibonacci Search
- Jump Search