1. First of all, Time complexity is not time taken by the program to execute. Time taken by the program to run varies on the system. For example , If I run a program on an old machine and run the same program on a new machine like a mackbook then program will take less time to execute on the new machine and so it is not the correct criteria to analyze an algorithm.

   So, instead we judge an algo based on how many computations it has to do to achieve our desired output. So if less computations are required then it is a more optimized algoritm.

   Therefore, TC is not calculated in units of time but in Big O notations which consists of a mathematical representation depending on the size of input.

2. Arrays are use to store one type of data but it stores it statically whereas Linked list stores data dynamically. Storing data statically means once the programmer defines the size of a data structure it can not be increased or decreased at run time as per the user's requirement. Whereas, if we store data dynamically then we can add and remove any data at run time.

   One more advantage of linked list over array is that while implementing linked list we either user the concept of structure or class and because of it we can store different types of data in one linked list only which is not possible in the case of arrays.
3. When a function is called inside it own function, it is called recursion. Since before complete execution of the function it is again called therefore a stack is maintained in the memory to store all the function calls. There is always a base condition while defining the recursive functions keeping in mind till which extent this function will keep calling itself and when and which condition will end this recursive calling.

   Recursion can be used to reverse a linked list, implement binary search, merge sort.

4. Sorting algos can we useful as a pre requisite for some other alog like to apply binary search we need a sorted array.

   Example of a simple sorting algo is Bubble sort,in this algo we traverse the array while comparing two adjacent numbers and swap them when the first element is greater then the lateral. With each iteration, we can observe that the largest element among the unsorted subarray is moving to the last index thus sorting the array from the back.
   In worst case it takes O(n**2)
   Another popular algo is Merger sort, It falls under divided and conquer algo. The basic idea is we will hypothetically divide the array into subarrays till this sub array gets sorted. (array is divided till a single element, and a single element is always sorted.) Then we merge the two sorted sub arrays till we sorted the entire array.
   Its TC = O(nlogn)
5. I just know that these algo come under graph theory .I have not done them yet.

6. Hash table is keeping the record of all the frequencies of all the elements that are provide to us as an input in form of array or some other data structure. By maintaining a hash table we can very easily find out the frequency at any given point of time.

   Hash table can be useful when I want to limit the login attempts of a user accessing a sit.
7. I have not done Dynamic programming.
8. Stack follows the principle of last in-first out whereas queue follows first in first out.
   Queue can be used when a number of users are requesting a resource like in coding contests when you hit run then you are put in a queue with other users who also hit run and want their code to be tested. In this scenario the first user clicking run will be given preference and so first in and first out.
   Stack can be used to see the last transaction or while maintain the record of bills where the user preferable request for the last entry and so last in first out.
9. Binary search falls under divide and conquer algo. Its pre requisite is that the elements must be sorted. The basic idea(explanation through array) is that we take the middle element of the array and compare it with the element we want to find, if our element is bigger than middle then we neglet the left side of the middle element and again starting searching in the new sub-array which starts from middle as starting index of the new sub array. Again find a new middle and similarly form a new sub array till our element is equal to the middle element.

   Its advantage over linear search is that it greatly reduces the time complexity. Linear search (in worst case) take O(n) whereas Binary search takes O(logn). This gap widens as the size of the array is increased.

10. I have not studied Binary tree till now.