№1

1. An algorithm is a step-by-step procedure or set of rules for solving a problem or completing a task. It is a sequence of well-defined instructions that take some input and produce an output.

2. When people refer to a "computer algorithm," they are talking about a specific set of instructions or rules that a computer program follows in order to solve a problem or perform a task. It is a precise and unambiguous description of how to solve a problem, designed to be implemented on a computer.

3. Algorithms can be used to solve a wide range of problems, including mathematical calculations, data sorting and searching, image processing, network routing, artificial intelligence, and many more. They are used in various fields such as computer science, mathematics, engineering, and finance.

4. Implementation of simple versions of basic algorithms can help in understanding the fundamental concepts of algorithms and their working principles. It can also improve problem-solving skills and logical thinking abilities. Additionally, implementing algorithms can provide hands-on experience in programming and computational thinking.

5. Sorting algorithms include:

- Bubble sort: It compares adjacent elements and swaps them if they are in the wrong order. It repeats this process until the list is sorted. Bubble sort can be used for small lists or when simplicity is more important than efficiency.

- Insertion sort: It builds the final sorted array one item at a time by comparing each new element with the already sorted elements and inserting it into the correct position. Insertion sort is efficient for small lists or partially sorted lists.

- Quicksort: It uses a divide-and-conquer approach to sort elements by selecting a pivot element, partitioning the array into two sub-arrays based on the pivot, and recursively sorting the sub-arrays. Quicksort is efficient for large lists and has an average-case time complexity of O(n log n).

6. Searching methods involve finding a specific element or value within a collection of data. Some common searching methods include linear search, binary search, and hash-based search algorithms.

- Linear search: It checks each element in a list or array sequentially until the desired element is found or the end of the list is reached. Linear search is simple but can be inefficient for large lists.

- Binary search: It is used for sorted lists or arrays and divides the search space in half by comparing the middle element with the desired value. It repeats this process until the element is found or the search space is empty. Binary search has a time complexity of O(log n) and is efficient for large sorted lists.

- Hash-based search algorithms: These algorithms use a hash function to map keys to array indices, allowing for efficient retrieval of values based on their keys. Hash-based search algorithms are commonly used in data structures like hash tables.

№2

1) General-purpose methods: These are methods or algorithms that are designed to be applicable to a wide range of problems or tasks, rather than being specific to a particular problem domain. They can be used in various contexts and are not limited to a specific application or field.

2) Heap data structure: It is a complete binary tree-based data structure that satisfies the heap property. The heap property states that for any given node, the value of the parent node is either greater than or equal to (in a max heap) or less than or equal to (in a min heap) the values of its children nodes. Heaps are commonly used in priority queues and sorting algorithms.

3) Divide-and-conquer algorithm: It is a problem-solving technique that involves breaking down a problem into smaller subproblems, solving each subproblem independently, and then combining the solutions to solve the original problem. This approach simplifies complex problems by dividing them into more manageable parts.

4) Insertion sort: It is a simple sorting algorithm that builds the final sorted array one item at a time. It iterates through the list, comparing each new element with the already sorted elements and inserting it into the correct position. Insertion sort has a time complexity of O(n^2) but can be efficient for small lists or partially sorted lists.

5) Adjacent elements: In the context of algorithms, adjacent elements refer to elements in a sequence or array that are next to each other in order. For example, in an array [1, 2, 3, 4], the adjacent elements are 1 and 2, 2 and 3, and 3 and 4.

6) Shell sort: It is a variation of insertion sort that improves its efficiency by comparing and swapping elements that are far apart before performing the final insertion sort. Shell sort uses a sequence of increment values to determine the gap between elements to be compared and swapped. It has an average-case time complexity of O(n log n) and is efficient for medium-sized lists.

7) Bubble sort: It is a simple sorting algorithm that repeatedly compares adjacent elements and swaps them if they are in the wrong order. This process is repeated until the list is sorted. Bubble sort has a time complexity of O(n^2) and is commonly used for small lists or when simplicity is more important than efficiency.

8) Quicksort: It is a divide-and-conquer sorting algorithm that selects a pivot element, partitions the array into two sub-arrays based on the pivot, and recursively sorts the sub-arrays. Quicksort has an average-case time complexity of O(n log n) and is efficient for large lists.

9) String processing: It refers to the manipulation and analysis of text or character data. String processing algorithms involve tasks such as searching for patterns, extracting substrings, replacing characters, and performing various operations on strings.

10) Variable-length coding: It is a technique used in data compression to represent symbols or data with varying lengths of code words. It assigns shorter codes to more frequently occurring symbols and longer codes to less frequently occurring symbols, resulting in overall compression of the data.

11) A message digest: It is a fixed-size numeric representation (usually a hash value) of a message or data. It is generated by applying a cryptographic hash function to the input data, and it is used to verify the integrity and authenticity of the message. A message digest is commonly used in digital signatures and checksums.

№3

1. relinquish - to give up or surrender

2. raison d'être - the most important reason or purpose for someone or something's existence

3. polynomial - an algebraic expression consisting of several terms

4. hale - healthy and strong

5. sequential - following a logical order or sequence

6. parsing - the process of analyzing a text to determine its grammatical structure

7. homogeneous - consisting of elements or parts that are all of the same kind or nature

8. multidimensional - having or involving multiple dimensions or aspects

№5

1. A programming algorithm is like: a recipe

2. An algorithm is written down as: a list of steps

3. An algorithm always leads to a(n): solution