**what is unit testing and out system and block box and white box?**

**Unit Testing:**

**Definition: Unit testing is the process of testing individual units or components of a software application in isolation. A "unit" is the smallest testable part of an application, such as a function or method.**

**Focus: The primary goal is to ensure that each unit of the software performs as designed. It helps in identifying and fixing bugs at an early stage of the development process.**

**Black Box Testing:**

**Definition: Black box testing is a testing method where the tester does not have any knowledge of the internal workings or code structure of the software being tested.**

**Focus: The focus is on the inputs and expected outputs of the software under test. Testers are concerned with the functionality of the system, not how that functionality is implemented.**

**White Box Testing:**

**Definition: White box testing, also known as glass box or clear box testing, involves testing the internal structure or workings of a software application. The tester has knowledge of the code, algorithms, and data structures.**

**Focus: The focus is on ensuring that all lines of code are executed and tested. It aims to verify the logic, flow, and internal structures of the software.**

**System Testing:**

**Definition: System testing is the process of testing the entire software system as a whole. It involves testing the integrated system to verify that it meets specified requirements.**

**Focus: The focus is on validating the overall system functionality, performance, and behavior. It ensures that all components work together as intended.**

**Different code injection and sql injection and dependence injection?**

**Code Injection:**

**Definition: Code injection refers to the act of introducing code into a software system. This can occur when an attacker is able to insert or manipulate code in a way that was not intended by the application's designers.**

**Examples: Common types of code injection include shell injection, where malicious commands are injected into a system shell, and HTML injection, where malicious HTML or script code is injected into a web page.**

**SQL Injection:**

**Definition: SQL injection is a specific type of code injection where an attacker inserts malicious SQL code into a query. This often occurs when user input is not properly sanitized or validated before being incorporated into a SQL query.**

**Example: If a web application takes user input for a login form and constructs a SQL query without proper validation, an attacker could input something like '; DROP TABLE users; -- as a username, which might manipulate the SQL query to delete the entire "users" table.**

**Dependency Injection:**

**Definition: Dependency injection is a design pattern used in software development where the dependencies of a component (such as a class) are injected from the outside rather than being created or managed within the component itself.**

**Purpose: Dependency injection is commonly used to achieve inversion of control, making components more modular, testable, and flexible. It helps in managing the dependencies between different components of a system.**

**Example: Instead of a class creating its own instances of dependent classes, those instances are provided (injected) from the outside. This can be done through constructor injection, method injection, or property injection.**

**complexity for all Data Structure?**

1. **Array:-**
   1. **Access O(1)**
   2. **Insert at end O(1)**
   3. **Delete from end O(1)**
   4. **Search O(n)**
   5. **Insert at middle O(n)**
   6. **Delete at middle O(n)**
   7. **Space Complexity O(n)**
2. **LinkedList:-**
   1. **Access O(n)**
   2. **Insert at begin O(1)**
   3. **Delete from begin O(1)**
   4. **Search O(n)**
   5. **Insert at middle O(n)**
   6. **Delete at middle O(n)**
   7. **Space Complexity O(n)**
3. **Stack**
   1. **Push O(1)**
   2. **Pop O(1)**
   3. **Space Complexity O(n)**
4. **Queue**
   1. **Enqueue O(1)**
   2. **Dequeue O(1)**
5. **Tree (BST,AVL)**
   1. **Search, insertion, delete O(log n) in balanced tree**
   2. **Space Complexity O(n)**
6. **Heap**
   1. **Heapfiy (Build Heap) O(n)**
   2. **Insertion O(log n)**
   3. **Space Complexity O(n)**
7. **Hash** 
   1. **Search: O(1) on average (O(n) in worst cases)**
   2. **Insertion: O(1) on average (O(n) in worst cases)**
   3. **Deletion: O(1) on average (O(n) in worst cases)**
   4. **Space Complexity O(n) in worst case**
8. **Graph**
   1. **Search: O(V + E)**
   2. **Insertion: O(1)**
   3. **Deletion: O(1)**
   4. **Space Complexity: O(V + E)**

**where and what can use multimap and give example?**

**multimap is a container in C++ Standard Template Library (STL) that is similar to map, but it allows multiple elements with the same key. It is implemented as a sorted associative container, meaning the elements are sorted according to their keys.**

**You can use multimap in scenarios where you need to associate multiple values with the same key.**

**# Create a dictionary with int as the key type and a list of strings as the value type**

**my\_multimap = {}**

**# Insert some values with the same key**

**my\_multimap.setdefault(1, []).append("Apple")**

**my\_multimap.setdefault(2, []).append("Banana")**

**my\_multimap.setdefault(1, []).append("Apricot")**

**my\_multimap.setdefault(3, []).append("Cherry")**

**my\_multimap.setdefault(2, []).append("Blueberry")**

**# Print the multimap**

**print("Multimap contents:")**

**for key, values in my\_multimap.items():**

**for value in values:**

**print(f"Key: {key}, Value: {value}")**

**# Accessing elements with a specific key**

**key\_to\_find = 2**

**values\_with\_key = my\_multimap.get(key\_to\_find, [])**

**print(f"\nValues with key {key\_to\_find}:")**

**for value in values\_with\_key:**

**print(f"Value: {value}")**

**using multimap in real life:-**

**Event Scheduling:**

**In a calendar or scheduling application, events scheduled for the same time can be stored in a multimap, where the key is the timestamp, and the values are the events.**

**Log Files:**

**When logging events in a system, timestamps or event types could be used as keys in a multimap, with multiple log entries associated with each key.**

**Network Programming:**

**In networking, a server might need to handle multiple connections at the same time. A multimap could be used to associate socket descriptors or connection identifiers with data related to each connection.**

**Database Indexing:**

**In database indexing, a multimap-like structure can be useful when indexing on non-unique keys. For example, an index on a person's last name might have multiple entries for each last name.**

**Flight Schedules:**

**In an airline reservation system, flight schedules could be stored in a multimap where the key is the destination airport code and the values are the scheduled departure times.**

**Social Networks:**

**Social networks often deal with relationships between users. A multimap could be used to store friendships or associations between users where a user ID is the key, and the values are the IDs of friends or connections.**

**Search Engines:**

**In a search engine, a multimap could be used to store search results where the search query is the key, and the values are the URLs or documents related to that query.**

**using Hash map in real life?**

**Database Indexing:**

**Hashmaps are used in database indexing to speed up the retrieval of records. For instance, an index on a user ID or a product ID can be implemented using a hashmap, allowing quick access to the associated data.**

**Caching:**

**In web development, hashmaps are frequently employed in caching mechanisms. For instance, a cache of recently accessed web pages or database query results can be implemented using a hashmap for quick retrieval.**

**Language Processing:**

**Spell checkers and autocomplete features in word processors or search engines often use hashmaps to store a dictionary or a list of frequently used words for efficient and quick lookups.**

**Phone Contacts:**

**Contact lists in smartphones use hashmaps to associate names with phone numbers. This enables fast lookups when searching for a specific contact.**

**File Systems:**

**Hashmaps are used in file systems to implement directory structures. The hashmap allows efficient mapping of file names to their corresponding file blocks or addresses on disk.**

**Symbol Tables in Compilers:**

**Compilers use hashmaps to implement symbol tables, which map variable and function names to memory addresses or other relevant information during the compilation process.**

**Network Routing Tables:**

**Hashmaps can be used in network routers to store routing tables. IP addresses can be hashed to determine the next hop for routing a packet efficiently.**

**Frequency Counting:**

**Hashmaps are often used to count the frequency of words in a document or the occurrence of items in a dataset, which is useful in natural language processing, data mining, and analytics.**

**User Authentication:**

**Hashmaps can be used to store user credentials (username and hashed passwords) for efficient and secure authentication processes.**

**Game Development:**

**In game development, hashmaps are utilized for quick access to game assets, such as textures, models, or sound files, based on unique identifiers.**

**what the library is used to using Graph in python?**

**networks :-** **the networkx library is commonly used for working with graphs. networkx is a powerful and flexible library for the creation, manipulation, and analysis of complex networks or graphs. It provides a wide range of functionality for graph algorithms, visualization, and graph-based data analysis.**

**igraph:- is a versatile library that provides a high-performance graph implementation and supports various graph algorithms and visualization capabilities. It's particularly useful for large-scale graph analysis.**

**example for using Binary search tree?**

**File Systems:**

**File systems often use BSTs or variants like AVL trees for quick lookups of file names. The hierarchical structure of directories can be efficiently represented using a BST.**

**Databases:**

**Many databases use BSTs as part of their indexing structures. For example, in a database table, the primary key index might be implemented as a BST, allowing for fast searches, insertions, and deletions based on the primary key.**

**Router Tables in Networking:**

**In networking, BSTs are used in the construction of router tables. IP addresses can be organized in a BST to facilitate fast lookups for routing decisions.**

**Library Systems:**

**Library systems can use BSTs to organize books based on their unique identifiers (e.g., ISBN numbers). This facilitates efficient searching for books and managing the library inventory.**

**Online Maps and Geographical Information Systems (GIS):**

**Maps and GIS applications often use BSTs for storing geographical locations and related information. This allows for efficient spatial queries.**

**Priority Queues:**

**Priority queues implemented using BSTs are useful in scenarios where elements have priorities, such as task scheduling in operating systems.**

**Organizing Hierarchical Data:**

**BSTs are useful for organizing hierarchical data in general. For example, representing organizational charts or file directory structures.**

**Database Query Optimization:**

**In database query optimization, BSTs can be used to optimize the search for rows based on indexed columns, improving the efficiency of queries.**

**example for using Red Black Tree tree?**

**File System Structure:**

**Imagine you are designing a file system where each file or directory has a unique identifier, such as a file path or name. A Red-Black Tree could be used to efficiently organize and manage the file system's hierarchical structure.**

**from PriorityQueue print from Bigger to smaller?**

**Two ways :**

1. **Give number with (-)sign**
2. **Store this in list and reversed**