Part-9

Sorting Algorithms in Python

Bubble Sort

- Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and sw aps them if they are in the wrong order.
- The pass through the list is repeated until the list is sorted.
- The algorithm gets its name from the way smaller or larger elements "bubble" to the top of the list.

Key Points:

- Compares adjacent elements and swaps them if they are in the wrong order.
- Passes through the list repeatedly until it is sorted.
- Smaller/larger elements "bubble" to the top of the list.

Map, Filter, and Reduce Functions

- **Map:** Applies a function of your choice to each item in an iterable (e.g., a list) and returns a new iterable with the tran sformed items.
- **Filter:** Creates a new iterable (e.g., a list) with the elements that pass the test implemented by the function you provi de.
- **Reduce:** Applies a function of two arguments cumulatively to the elements of a sequence, from left to right, to reduc e the sequence to a single value.

Key Points:

- Map applies a function to each item in an iterable.
- Filter creates a new iterable with elements that pass a given test.
- Reduce applies a function cumulatively to the elements of a sequence.

Exception Handling

- **Exception Handling:** The process of handling runtime errors that terminate the execution of a program.
- **Types of Errors:**
  - Compile-time errors: Errors detected by the compiler, such as syntax or indentation errors.
  - Logical errors: Errors in the program's logic that result in unexpected output.
  - Runtime errors: Errors that occur during the execution of the program, such as division by zero or accessing an und efined variable.
- **Exception Handling Syntax:**
  - `try`: Encloses the code that might raise an exception.
  - `except`: Handles the exception if it occurs.
  - `else`: Executes if no exception occurs in the `try` block.
  - `finally`: Executes regardless of whether an exception occurred or not.

  Key Points:
- Exception handling allows you to handle runtime errors and provide meaningful error messages.
- There are three main types of errors: compile-time, logical, and runtime.
- The exception handling syntax uses `try`, `except`, `else`, and `finally` blocks to handle exceptions.

File Handling

- **File Handling:** The process of creating, reading, updating, and deleting files in Python.
- **File Handling Process:**
  1. Open a file using the `open()` function.
  2. Perform read, write, or append operations on the file.
  3. Close the file.
- **File Modes:**
  - `r`(read): Opens the file for reading (default mode).
  - `w`(write): Opens the file for writing, creating a new file if it doesn't exist.
  - `a`(append): Opens the file for appending, creating a new file if it doesn't exist.
  - `x`(create): Creates a new file, returns an error if the file exists.
  - `t`(text): Opens the file in text mode (default).
  - `b`(binary): Opens the file in binary mode.
- **Advantages of File Handling:**

- Data is stored permanently.
- Data can be shared with others.
- Data can be updated.

Key Points:

- File handling involves opening, reading, writing, and closing files.
- There are various file modes, such as read, write, append, and create.
- Files can be opened in text or binary mode.
- File handling provides advantages like permanent data storage, data sharing, and data updating.

Sorting Algorithms

Bubble Sort

- Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and sw aps them if they are in the wrong order.
- The pass through the list is repeated until the list is sorted.

Insertion Sort

- Insertion sort places a given element at the right position in a sorted list.
- The list is virtually split into a sorted and an unsorted part.
- Values from the unsorted part are picked and placed at the correct position in the sorted part.

Merge Sort

- Merge sort is a divide-and-conquer algorithm.
- It first divides the list into equal halves, then recursively sorts the halves, and finally merges the sorted halves.

Quick Sort

- Quick sort is also a divide-and-conquer algorithm.
- It chooses a 'pivot' element from the array and partitions the other elements into two sub-arrays, according to whet her they are less than or greater than the pivot.
- The sub-arrays are then recursively sorted.

Key Points:

- Bubble sort: Compares adjacent elements and swaps them if they are in the wrong order.
- Insertion sort: Places a given element at the right position in a sorted list.
- Merge sort: Divides the list, recursively sorts the halves, and merges the sorted halves.
- Quick sort: Chooses a 'pivot' element and partitions the other elements into two sub-arrays.

Algorithm	**Time Complexity (Average)**	**Time Complexity (Worst)**	**Space Complexity**
Bubble Sort	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n^2)	O(n^2)	O(1)
Merge Sort	O(n log n)	O(n log n)	O(n)
Quick Sort	O(n log n)	O(n^2)	O(log n)