**LAR Liyanage**

**27336**

**Tutorial 05**

1. Explain how main sorting algorithms can be performed using an appropriate array example.

* Bubble Sort :- Bubble sort is a simple comparison-based sorting algorithm. It repeatedly swaps adjacent elements if they are in the wrong order until the entire array is sorted.

Let's consider an example array: [5, 2, 8, 12, 1]

* Pass 1: [2, 5, 8, 1, 12]
* Pass 2: [2, 5, 1, 8, 12]
* Pass 3: [2, 1, 5, 8, 12]
* Pass 4: [1, 2, 5, 8, 12]
* Selection Sort :- Selection sort is another simple comparison-based sorting algorithm. It repeatedly finds the minimum element from the unsorted part of the array and puts it at the beginning.

Let's consider the same example array: [5, 2, 8, 12, 1]

* Pass 1: [1, 2, 8, 12, 5]
* Pass 2: [1, 2, 8, 12, 5]
* Pass 3: [1, 2, 5, 12, 8]
* Pass 4: [1, 2, 5, 8, 12]
* Insertion Sort :- Insertion sort is a simple comparison-based sorting algorithm. It builds the final sorted array one element at a time, by repeatedly inserting the next element into the sorted part of the array.

Let's consider the same example array: [5, 2, 8, 12, 1]

* Pass 1: [2, 5, 8, 12, 1]
* Pass 2: [2, 5, 8, 12, 1]
* Pass 3: [2, 5, 8, 12, 1]
* Pass 4: [1, 2, 5, 8, 12]
* Merge Sort :- Merge sort is a divide-and-con

1. Compare and contrast bubble sort and selection sort algorithms.

* Bubble Sort.
  + Bubble sort works by repeatedly swapping adjacent elements if they are in the wrong order until the entire list is sorted.
  + It has a time complexity of O(n^2) in the average and worst cases, where n is the number of elements in the list.
  + Bubble sort is an in-place sorting algorithm, meaning it doesn't require additional memory to perform the sorting.
  + It is relatively simple to understand and implement, making it a good choice for small lists or educational purposes.
  + Bubble sort is not well-suited for large lists or lists that are mostly sorted, as it has poor performance in these cases.
  + It is a stable sorting algorithm, meaning elements with equal values retain their relative order after sorting.
* Selection Sort.
  + Selection sort works by repeatedly finding the minimum element from the unsorted part of the list and swapping it with the element at the beginning of the unsorted part until the entire list is sorted.
  + It also has a time complexity of O(n^2) in the average and worst cases.
  + Like bubble sort, selection sort is an in-place sorting algorithm that doesn't require additional memory.
  + Selection sort performs better than bubble sort in practice, as it reduces the number of swaps required compared to bubble sort.
  + Similar to bubble sort, selection sort is not efficient for large lists or mostly sorted lists.
  + It is also a stable sorting algorithm.
  + Both algorithms are easy to understand and implement but are not recommended for large or mostly sorted lists.

1. What are the real-world examples of sorting?

* Contacts List :- When you sort the contacts list on your phone alphabetically by name, you are using a sorting algorithm. This allows you to quickly find and access the contact you're looking for.
* Library Catalog :- Libraries use sorting algorithms to organize books by author, title, or subject. This helps users easily locate books based on their search criteria.
* E-commerce Platforms :- E-commerce platforms use sorting algorithms to sort products based on relevance, price, popularity, customer ratings, or other factors. This allows customers to view products in the desired order, making it easier to find what they are looking for.
* Search Engine Results :- Search engines use sorting algorithms to rank search results based on their relevance to the search query. Sorting ensures that the most relevant and useful results are displayed at the top of the search results page.
* Playlist Sorting :- Music streaming platforms allow users to create playlists and sort the songs within them. Sorting options can include sorting by title, artist, genre, or duration. This helps users organize their music collection according to their preferences.
* Flight Schedules :- Airlines use sorting algorithms to arrange flight schedules by departure time, arrival time, or flight number. This ensures that flights are displayed in a logical and easily navigable manner for travelers.
* Racing Rankings :- In sports events like marathons or car races, the final rankings are determined by sorting the participants based on their finishing times. This helps determine the winners and provides an organized display of the results.

1. Write a function using pseudo or source codes to sort an integer array using bubble sort and selection sort.

* Bubble Sort.

def bubble\_sort(arr):

n = len(arr)

for i in range(n-1):

for j in range(0, n-i-1):

if arr[j] > arr[j+1] :

arr[j], arr[j+1] = arr[j+1], arr[j]

return arr

* Selection Sort.

def selection\_sort(arr):

n = len(arr)

for i in range(n):

min\_idx = i

for j in range(i+1, n):

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

return arr