Date: 28 06 2023

Sorting Algorithms.

Tut.09

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

Selection sort -: The algorithm repeatedly selects the smallest / largest element from the unsorted portion of the list and swaps it with the first element of the unsorted part. This process is continued until the to entire list is sorted.

Eg :- arr [ ] = { 64 , 25 , 12 , 22 , 11 } .

Bubble sort -: This is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

Eg :- how the contact list on your phone is sorted in alphabetical order.

2) Real world ~~expl~~ example for sorting algorithms.

~~Selection sort~~ :-

Traffic lights

Sorting the contactlists

Sorting from max to min prices in online shopping pages

3) Compare and contrast bubble sort and selection sort algorithm

Bubble sort	Selection Sort
Less efficient	More efficient
User item exchanging	User item selection
Slower	Faster
Compare the adjacent elements and swap accordingly	Selects the minimum from the unsorted subarray and place them in the sorted array.
Compare the adajcent pair to sort	Takes the smallest value in the list and move it to the proper position in the array

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

```
# include <stdio.h>
void bubbleSort(int array[].int size)
{
  for (int step = 0; step < size - 1; ++step)
{ {
   for (int i = 0; i < size-step = 1; ++i)
   {
    if (array[i] > array[i = 1])
    {
      int temp = array[i];
      array[i] = array[i - 1];
      array[i + 1] = temp;
    }
   }
  }
}

void printArray(int array[], int size) {
  for (int i = 0; i < size; ++i) {
    printf("%d ", array[i]);
  }
  printf("\n");
}
```

```
void selectionsort (int array [], int size)
 {
  for (int step = 0; step <size -1; step++)
   {
     int min_idx = step;

     for (int i = step +1; i< size; i++)
      {
       if (array [i] < array [min_idx]

       min_idx = i;
      }
     swap (& array [min_idx], & array [step]);
     }
   }

void printArray (int Array [], int size)
 {
   for (int i = 0; i<size; ++i)
   {
     printf ("%d", array [i]);
   }
   printf ("\n");
 }
```