Kartikeya Kumar
Panel - E
Roll no. - 38
Batch: E2
PRN: 10322320023

Page No. 01
Date 06 08 2024

DATA STRUCTURES

9/8/24

EXPERIMENT NO. - 1 SORTING

PROBLEM STATEMENT

Write a C program to create a student database using an array of structures. Apply sorting techniques (Bubble sort, Insertion sort and Selection sort)

OBJECTIVE

1. To understand the use of an array of structures for maintaining records.
2. To implement and analyze sorting techniques.
3. To compare bubble sort, insertion sort and selection sort.

IMPLEMENTATION

☆ Platform

→ 64-bit Open source Linux or its derivatives.
→ Open source C programming tool like gcc / Eclipse Editor.

☆ Test Conditions

1. Input at least five records.
2. Arrange list of students roll call listwise. (Use Bubble sort)
3. Arrange list of students rollcall listwise (@Use Insertion sort and Selection sort)
4. Test the three sorting methods for at least five records.

★ Pseudo Code

- **Bubble Sort:**

```
algorithm bubbleSort(a[], n)
{
    for i=1 to n
        for j=0 to n-i-1
            if a[j] > a[j+1]
            {   swap(a[j], a[j+1])
                temp = a[j]
                a[j] = a[j+1]
                a[j+1] = temp
            }
}
```

- **Selection Sort:**

```
algorithm selectionSort(a[], n)
{
    for i=0 to n-2
    {
        minpos = i
        for j=i+1 to n-1
            if a[j] < a[minpos]
                minpos = j
        if minpos != i
        {   temp = a[i]
            a[i] = a[minpos]
            a[minpos] = temp
        }
    }
}
```

- Insertion Sort :

```
algorithm insertionSort ( arr[], n)
{
    for i = 1 to n
    {
        key = arr[i]
        j = i - 1
        while j >= 0 and arr[j] > key
        {
            arr[j+1] = arr[j]
            j--
        }
        arr[j+1] = key
    }
}
```

☆ Time Complexity

- Bubble Sort :
  $f(n) = O(n)$ → Best case (already sorted array)
  $f(n) = O(n^2)$ → Average & worst case

- Selection Sort :
  $f(n) = O(n^2)$ → Best case (already sorted array)
  $f(n) = O(n^2)$ → Average to worst case

- Insertion Sort :
  $f(n) = O(n)$ → Best case (already sorted array)
  $f(n) = O(n^2)$ → Average to worst case

☆ Conclusion

Thus, implemented different sorting methods on the student database. This system is able to perform sorting under different cases.

☆ FAQs

1. What is the meaning of database? How to maintain it in C?

Ans. A database is an organized collection of data stored in a computer system. It is designed to efficiently store, retrieve and manage information.

While it is technically possible to build a database in C, it is generally not recommended due to ~~proceed~~ the complexities and inefficiencies involved.

We can use structures in C to define records of the database and use either in-memory or file-based storage, using file operations. We can develop functions for inserting, deleting, updating, retreiving ~~or~~ and sorting the records.

2. What are the applications of sorting?

Ans. Sorting has many applications such as:

- Database indexing
- File system organization
- ~~Search~~ Search algorithms, merge algorithms etc. (for faster execution)
- Finding medians, percentiles and quartiles (for efficient calculation)

- Data visualization - easier to represent sorted data in graphs and charts
- Machine learning - sorting is a preprocessing step for many algorithms
- Operating system - process scheduling, memory management, file systems
- Networking - routing algorithms, packet handling
- Compiler opmization
- Cryptography - some algorithms rely on sorting

3. Compare and contrast bubble, selection, insertion and shell sort.

Ans. Bubble Sort: Compares adjacent elements and swaps them if needed. It is less efficient as compared with other sorting techniques.
Time complexity is $O(n^2)$

Selection Sort: It finds minimum element and swaps with the first element in each iteration.
It is slightly more efficient than bubble sort with a time complexity of $O(n^2)$

Insertion Sort: It builds the sorted array one element at a time by inserting elements into their correct positions. It is efficient for small datasets or nearly sorted arrays. Time complexity is $O(n^2)$

Shell sort: It is an improved version of insertion sort with better time complexity of $O(n \log n)$ in average cases.
It divides the array into smaller subarrays, sorts them using insertion sort and gradually reduces gap between elements. Most efficient out of the four techniques.

4. What is external and internal sorting and in-place and stable features of sorting algorithms?

Ans. Internal sorting: All the data is stored in main memory at all times while sorting. It is used when data is small enough to fit in main memory. e.g., Bubble sort, insertion sort, quick sort, heap sort etc.

External sorting: The data is stored outside memory (like on disk) and only loaded into memory in small chunks. It is used when data cannot fit into main memory entirely. e.g., external merge sort etc.

In-place sorting: Sorting is done without using extra space proportional to input size. It modifies input array directly.
e.g., bubble sort, insertion sort, selection sort, heap sort etc.

Stable sorting: A stable sort keeps equal elements in the same order as in input.
e.g., insertion sort, merge sort, bubble sort etc.