रा.इ.सू.प्रौ.सं NIELIT

NIELIT GORAKHPUR

Course Name: A Level (2nd Sem) **Subject: Data Struture using C++**
Topic: Types of Sorting in C++ **Date: 09-04-2020**

Sorting

Sorting involves arranging data in ascending or descending order, according to a certain collating sequence (or sorting sequence). The sorting algorithm includes:

- Insertion Sort
- Selection Sort
- Bubble Sort
- Merge Sort (Recursive Top-Down or Interactive Bottom-up)
- Quick Sort (Recursive)
- Bucket Sort
- Heap Sort
- Binary Tree Sort

.

Example1: Sorting an Array using Insertion Sort.

```
#include <iostream>
using namespace std;
void insertionSort(int a[], int size);
void print(const int a[], int iMin, int iMax);
int main() {
  const int SIZE = 8;
  int a[SIZE] = {8, 4, 5, 3, 2, 9, 4, 1};
  print(a, 0, SIZE - 1);
  cout << endl;
  insertionSort(a, SIZE);
  print(a, 0, SIZE - 1);
  cout << endl;
}
void insertionSort(int a[], int size)
{
  int temp;   // for shifting elements
  for (int i = 1; i < size; ++i)
  {
    print(a, 0, i - 1);                    // already sorted
    print(a, i, size - 1);                 // to be sorted
    cout << endl;

    for (int prev = 0; prev < i; ++prev)
   {
      if (a[i] < a[prev])
     {
        temp = a[i];
        for (int shift = i; shift > prev; --shift)
{
          a[shift] = a[shift-1];
        }
        a[prev] = temp;
        break;
      }
    }
  }
}
void print(const int a[], int iMin, int iMax)
{
  cout << "{";
  for (int i = iMin; i <= iMax; ++i)
  {
    cout << a[i];
    if (i < iMax) cout << ",";
  }
  cout << "} ";
}
```

Example2 : **Sorting an Array using Selection Sort.**

```
#include <iostream>
using namespace std;
void selectionSort(int a[], int size);
void print(const int a[], int iMin, int iMax);
int main()
{
  const int SIZE = 8;
  int a[SIZE] = {8, 4, 5, 3, 2, 9, 4, 1};
  print(a, 0, SIZE - 1);
  cout << endl;
  selectionSort(a, SIZE);
  print(a, 0, SIZE - 1);
  cout << endl;
}
void selectionSort(int a[], int size)
{
  int temp;                              // for swaping
  for (int i = 0; i < size - 1; ++i)
  {
    print(a, 0, i - 1);
    print(a, i, size - 1);
    int minIndex = i;                    // assume fist element is the smallest
    for (int j = i + 1; j < size; ++j)
    {
      if (a[j] < a[minIndex]) minIndex = j;
    }
    if (minIndex != i)
    {
      temp = a[i];
      a[i] = a[minIndex];
      a[minIndex] = temp;
    }
    cout << "=> ";
    print(a, 0, i - 1);
    print(a, i, size - 1);
    cout << endl;
  }
}
void print(const int a[], int iMin, int iMax)
{
  cout << "{";
  for (int i = iMin; i <= iMax; ++i)
  {
    cout << a[i];
    if (i < iMax) cout << ",";
  }
  cout << "} ";
}
```