**Experiment No. – 2.2**

**Aim**: Write a program to implement Shell sort along with its complexity analysis.

1. **Problem Description:**

We have given an unsorted array of numbers, generate a sorted array of numbers by applying Shell sort. Demonstrate knowledge of time complexity of Shell Sort and find space complexity.

1. **Algorithm:**

**Algorithm Shell\_Sort (DATA, N):**

ShellSort(a, n) // 'a' is the given array, 'n' is the size of array

for (interval = n/2; interval **>** 0; interval /= 2)

for ( i = interval; i **<** **n**; i += 1)

temp = a[i];

for (j = i; j **>**= interval && a[j - interval] **>** temp; j -= interval)

a[j] = a[j - interval];

a[j] = temp;

End ShellSort

1. **Complexity Analysis**

**Time complexity of Shell Sort**

* **Best Case Complexity -** **O(n\*logn).** It occurs when the array is already sorted.
* **Average Case Complexity -** **O(n\*logn).** It occurs when the array elements are in jumbled order that is not properly ascending and not properly descending.
* **Worst Case Complexity -** **O(n2).** It occurs when the array elements are required to be sorted in reverse order. That means suppose you have to sort the array elements in ascending order, but its elements are in descending order.

**Space Complexity**: O(1). As no extra space is used while sorting.

1. **Pseudo Code**

def shell\_sort(array):

gap = len(array) // 2

while gap > 0:

for i in range(gap, len(array)):

current\_item = array[i]

j = i

while j >= gap and array[j - gap] > current\_item:

array[j] = array[j - gap]

j -= gap

array[j] = current\_item

gap //= 2

1. **Source Code (C/C++):**

#include <bits/stdc++.h>

using namespace std;

void shellSort(int arr[], int n)

{

    int gap = n / 2;

    while (gap > 0)

    {

        for (int i = gap; i < n; i++)

        {

            int current = arr[i];

            int j = i;

            while (j >= gap && arr[j - gap] > current)

            {

                arr[j] = arr[j - gap];

                j -= gap;

            }

            arr[j] = current;

        }

        gap /= 2;

    }

}

void print(int arr[], int n)

{

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

}

int main()

{

    cout << "Name: Saurabh Kumar \nUID: 23MAI10004\n";

    cout << "\nShell Sort\n";

    int n;

    cout << "Enter Size of array\n";

    cin >> n;

    int arr[n];

    cout << "Enter array elements \n";

    for (int i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

    cout << "\nArray before sorting: ";

    print(arr, n);

    shellSort(arr, n);

    cout << "Array after sorting : ";

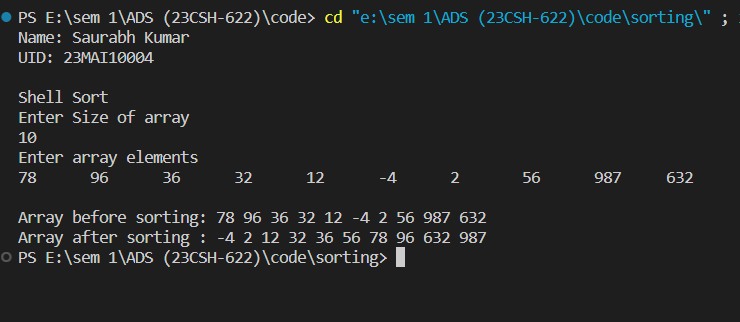
    print(arr, n);

    return 0;

    return 0;

}

1. **Screenshot of Outputs:**

****

1. **Learning Outcomes**
2. Learnt about how to sort an array.
3. Learnt about how to implement Shell sort.
4. Learnt about how to find time and space complexity
5. Learnt about how to code in c++ and take array as an input.
6. Learnt about how to use function and loop in c++.