Q1 f) Implement Shell Sort Algorithm. Analyze its time complexity.

TIME COMPLEXITY:
1. BEST CASE: O(n ^ 2)

SPACE COMPLEXITY: O(1)

```
#include <iostream>
#include <vector>
#include <random>
using namespace std;
void shellSort(vector<int> & arr)
{
    int n = int(arr.size());
    for (int gap = n / 2; gap > 0; gap /= 2)
        for (int i = gap; i < n; i += 1)
        {
            int temp = arr[i], j;
            for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)
                arr[j] = arr[j - gap];
            arr[j] = temp;
        }
}
int main()
{
    vector<int> A(100);
    for(int i = 0; i < 100; ++i)
        A[i] = rand() % 100;
    cout << "INITIAL STATE OF ARRAY:\n";
    for(int i = 0; i < 100; ++i)
        cout << A[i] << " ";
    cout << endl << endl;
    shellSort(A);
    cout << "SORTED ARRAY:\n";
    for(int i = 0; i < 100; ++i)
        cout << A[i] << " ";
    cout << endl;
}
```

OUTPUT:

```
INITIAL STATE OF ARRAY:
7 49 73 58 30 72 44 78 23 9 40 65 92 42 87 3 27 29 40 12 3 69 9 57 60 33 99
 78 16 35 97 26 12 67 10 33 79 49 79 21 67 72 93 36 85 45 28 91 94 57 1 53
 8 44 68 90 24 96 30 3 22 66 49 24 1 53 77 8 28 33 98 81 35 13 65 14 63 36
 25 69 15 94 29 1 17 95 5 4 51 98 88 23 5 82 52 66 16 37 38 44

SORTED ARRAY:
1 1 1 3 3 3 4 5 5 7 8 8 9 9 10 12 12 13 14 15 16 16 17 21 22 23 23 24 24 25
 26 27 28 28 29 29 30 30 33 33 33 35 35 36 36 37 38 40 40 42 44 44 44 45 49
 49 49 51 52 53 53 57 57 58 60 63 65 65 66 66 67 67 68 69 69 72 72 73 77 78
 78 79 79 81 82 85 87 88 90 91 92 93 94 94 95 96 97 98 98 99
Program ended with exit code: 0
```