APPLICATION OF BUBBLE SORTING ALGORITHM IN SWOT ANALYSIS

Samandarova Shaxnoza Mirjamilovna
(Karshi branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi department of Software of Information Technologies intern teacher.)
Axmedova Nilufar Farxodovna
(Karshi branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi department of Software of Information Technologies intern teacher.)
Mamatqulova Dildora Farxodovna
(Karshi branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi student.)

SWOT was originally created to analyze businesses in the corporate world. Its utility has now been made clear as a business tool that can be applied to many different situations; SWOT can be helpful for restaurant teams, non-profit structures, governmental bodies, and for individual quandaries on a person-to-person level.

Though seemingly self-explanatory, each of the four elements of a SWOT analysis chart have their own power in determining a situation.

- Strengths

➢ Strengths apply to what an organization excels at, where its talent lies, and what separates it from its competitors.

- Weaknesses

➢ Weaknesses are what prevents an organization from accessing those strengths or performing at its highest possible level.

- Opportunities

➢ Opportunities can refer to external factors that would enable an organization to develop a competitive advantage or sector of specialization that would distinguish them in their field.

- Threats

➢ Threats are both internal and external factors that could potentially harm an organization, affect its workflow or product output negatively, or hurt its workforce.

Bubble sort is a simple, inefficient sorting algorithm used to sort lists. It is generally one of the first algorithms taught in computer science courses because it is a good algorithm to learn to build intuition about sorting. While sorting is a simple concept, it is a basic principle used in complex computer programs such as file search, data compression, and path finding. Running time is an important thing to consider when selecting a sorting algorithm since efficiency is often thought of in terms of speed. Bubble sort has an average and worst-case running time of $O(n2)$, so in most cases, a faster algorithm is more desirable.

Let us understand the working of bubble sort with the help of the following illustration:

Int A[4]={6,3,0,5};

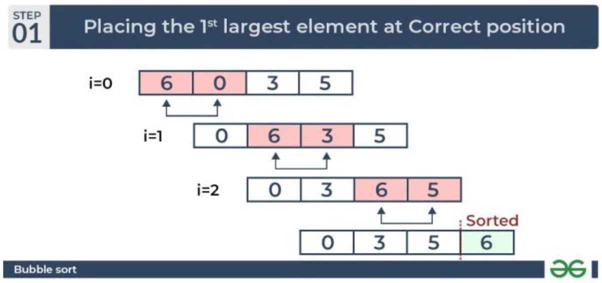


Fig. 1 First step

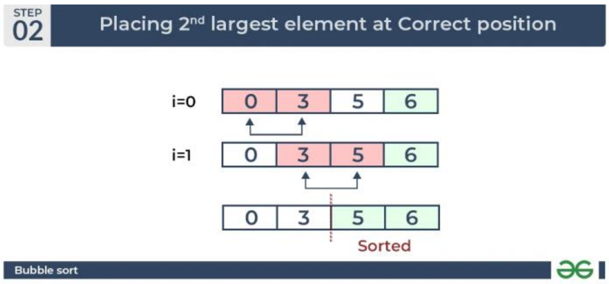


Fig. 2 Second step

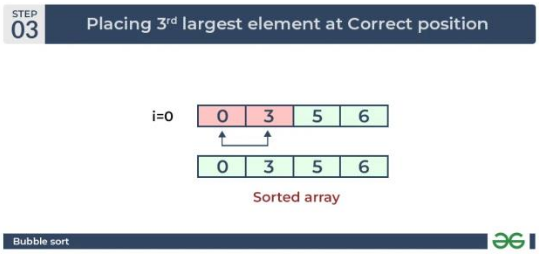


Fig. 3 Third step

Total no. of passes: n-1

Total no. of comparisons: n*(n-1)/2

Below is the implementation of the bubble sort. It can be optimized by stopping the algorithm if the inner loop didn't cause any swap.

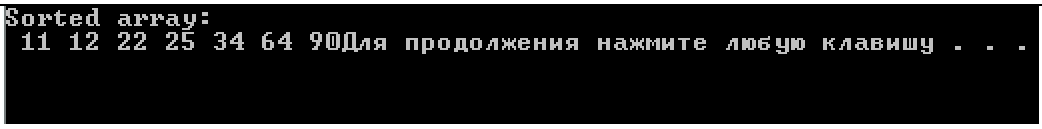
```
// Optimized implementation of Bubble sort
#include <iostream.h>
using namespace std;

// An optimized version of Bubble Sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    bool swapped;
    for (i = 0; i < n - 1; i++) {
            swapped = false;
            for (j = 0; j < n - i - 1; j++) {
                    if (arr[j] > arr[j + 1]) {
                            swap(arr[j], arr[j + 1]);
                            swapped = true;
                    }
            }

            // If no two elements were swapped
            // by inner loop, then break
            if (swapped == false)
                    break;
    }
}

// Function to print an array
void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
            cout << " " << arr[i];
}

// Driver program to test above functions
int main()
{
    int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
    int N = sizeof(arr) / sizeof(arr[0]);
    bubbleSort(arr, N);
    cout << "Sorted array: \n";
    printArray(arr, N);
    system("PAUSE");
    return 0;
}
// This code is contributed by shivanisinghss2110
```



```
Sorted array:
 11 12 22 25 34 64 90Для продолжения нажмите любую клавишу . . .
```

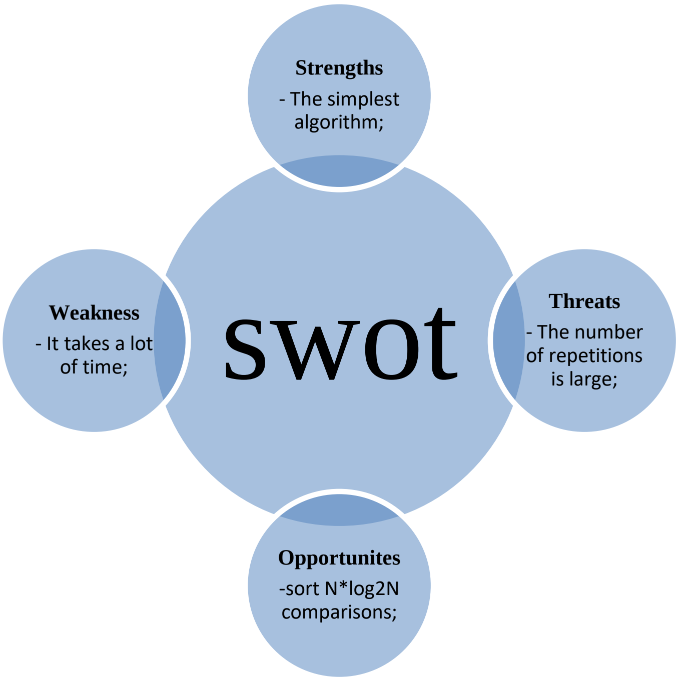

Strengths
- The simplest algorithm;
Weakness
- It takes a lot of time;
swot
Threats
- The number of repetitions is large;
Opportunites
-sort N*log2N comparisons;

Fig. 4 SWOT Analysis of Bubble Sort.

REFERENCES:

1. Алгоритмы: построение и анализ. 3-е изд. / Т.Х.Кормен.Ч.И.Лейзерсон, Р.Л.Ривест, К.Штайн. – М.: Вильямс, 2013. – 1328 с.

2. Вирт Н. Алгоритмы и структуры данных. //М., ДМК, 2010. – 272 с.

3. Даступа С., Пападимтриу Х., Вазирани У. Алгоритмы. – М.: МЦ-НМО, 2014 -320 с.

4. https://www.geeksforgeeks.org