Documentatie proiect

**Cuprins:**

1. **Specificaţii.**
2. **Conţinut**
3. **Referinţe**

**1.Specificaţiile programului:**

Proiectul ales: *“Se dă un tablou de N elemente aleatorii. Să se sorteze tabloul prin algoritmii Bubblesort, Shellsort și Quicksort și să se compare timpul de execuție pentru cele 3 metode.”*

**2.Continut:**

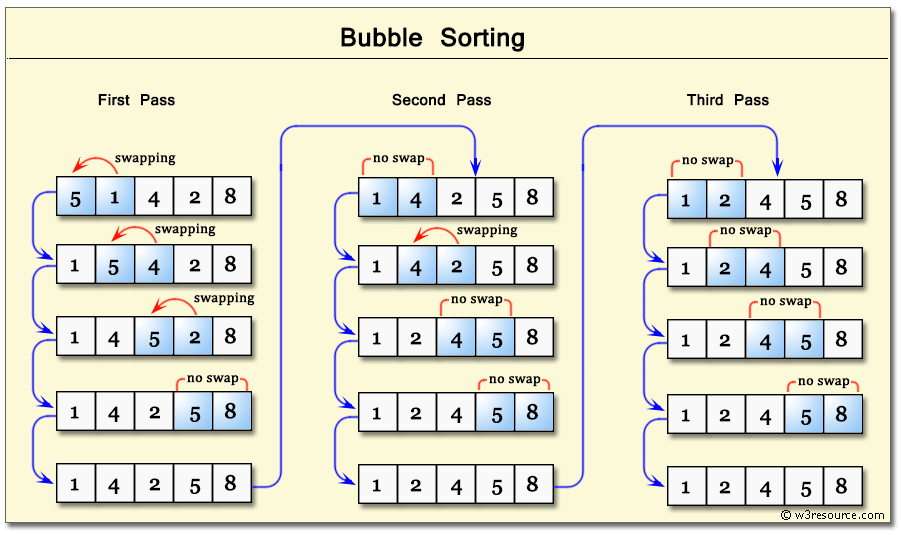
Initial, s-a implementat o functie de generare a unui vector aleator,functie realizata cu ajutorul functiilor rand si time din biblioteca *<time.h>*  a limbajului C.

Programul este bazat pe 3 functii distincte, cate una pentru fiecare tip de sortare.

* **BUBBLESORT:**

Este cel mai simplest mod de sortare a unui vector ce se bazeaza pe interschimbarea repetata a elementelor,daca acestea se afla in ordinea gresita.Sortarea compara cate 2 elemente, verificand care este mai mare si iterand pana ce vectorul este complet sortat. Nu este cea mai favorabila implementare pentru vectori de dimensiuni mai mari, dezavantajul fiind timpul de executie lent: de O(N²).

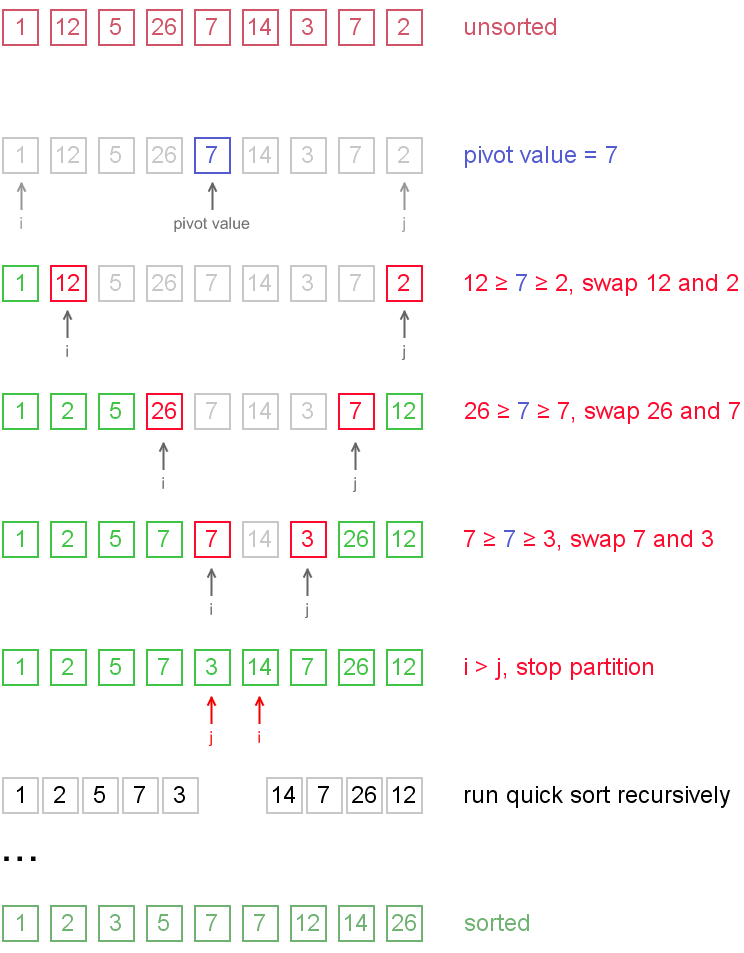
Exceptia este data de cazul cand vectorul este deja sortat si timpul de executie este de O(n).



* **QUICKSORT:**

**Quicksort** este un algoritm de tipul ***“Devide and Conquer”.*** Se bazeaza pe alegerea unui element ca pivot si se ordonează elementele listei, astfel încât toate elementele din stânga pivotului să fie mai mici sau egale cu acesta, și toate elementele din dreapta pivotului să fie mai mari sau egale cu acesta.

In algoritmul abordat in proiect pivotul este ales ca fiind mijlocul partitiei.



Pentru un set de n valori oarecare algoritmul efectuează O(nlogn) comparații, dar în cazul cel mai nefavorabil se efectuează O(n2) comparații.

Caz nefavorabil: cel mai rău caz apare atunci când se alege întotdeauna cel mai mare sau cel mai mic element ca pivot.

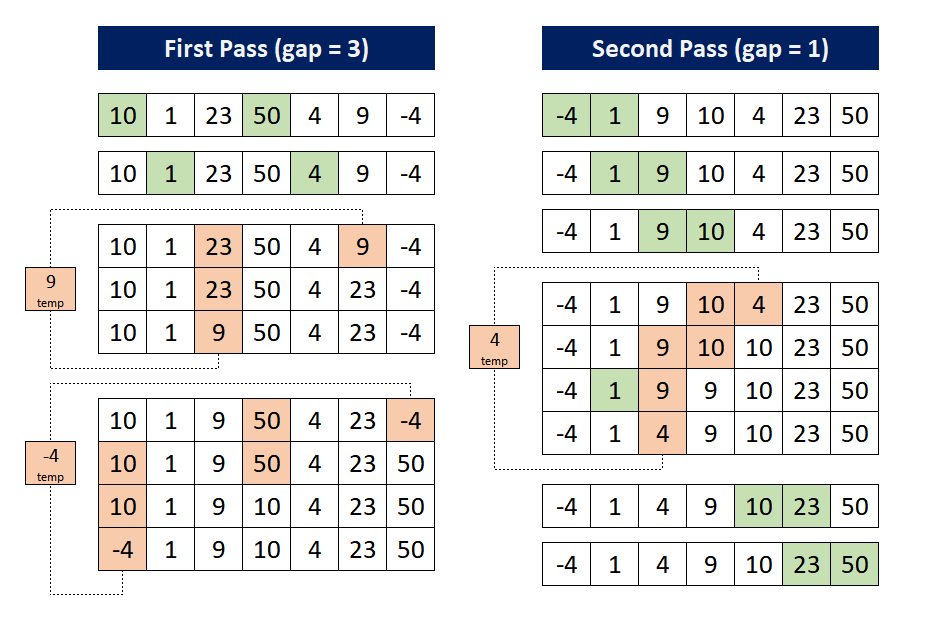
* **SHELLSORT:**

Algoritm de sortare performant,bazat pe sortarea prin insertie.Sortarea Shell este o versiune generalizată a algoritmului de sortare prin inserție. Mai întâi sortează elementele care sunt departe unele de altele și reduce succesiv intervalul dintre elementele de sortat. Intervalul dintre elemente este redus în funcție de secvența utilizată.

**Complexitatea timpului: este O(n2)..**

**Complexitatea celui mai rău caz:O(n2)**

**Complexitatea celui mai bun caz:** **Ω(n log(n))-**când lista de matrice dată este deja sortată, numărul total de comparații pentru fiecare interval este egal cu dimensiunea matricei date.



**3.Referinte:**

* [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)
* [**https://www.programiz.com/dsa/**](https://www.programiz.com/dsa/)