

Laborator 2

Se considera problema de la laboratorul 1 cu urmatoarea modificare:

Se cere un program care sa asigure urmatoarea postconditie:

Postconditie: Matricea initiala contine imaginea filtrata.

Constrangere: NU se alocă o alta matrice rezultat si nici o matrice temporara!

Se pot folosi/aloca doar vectori temporari pentru care complexitatea spatiu se incadreaza in $O(n)$.

Atentie si implementarea secventiala trebuie adaptata la constrangerea specificata.

Va fi nevoie sa se foloseasca vectori auxiliari temporari.

Obiectiv: optimizarea complexitatii-spatiu in conditiile obtinerii unei performante ridicate.

Datele de intrare se citesc dintr-un fisier de intrare "date.txt".

(Fisierul trebuie creat anterior prin adaugare de numere generate aleator.)

Rezultatul se salveaza intr-un fisier **output.txt**

Implementare

- a) Java
- b) C++ (cel putin C++11)

Fata de Laboratorul 1 se cere doar:

-Distributie pe linii

-Kernel de dimensiune $k=3$

-Pentru C++ alocare dinamica

Testare: masurati timpul de executie pentru

- 1) $N=M=10$ si $n=m=3$; $p=2$ +executie secventiala
- 2) $N=M=1000$ si $n=m=3$; $p=2,4,8,16$ +executie secventiala
- 3) $N=10000$ $M=10000$ si $n=m=3$; $p=2,4,8,16$ +executie secventiala

Observatii:

- Fiecare test trebuie repetat de 10 ori si evaluarea timpului de executie se considera media aritmetica a celor 10 rulari.
- Pentru fiecare executie (cele 10) a fiecarui caz de testare folositi acelasi fisier de intrare "date.txt" si verificati corectitudinea prin comparatie cu fisierul output obtinut la executia secventiala.

Analiza:

Comparati performanta pentru fiecare caz !

Comparati timpii de executie obtinuti cu implementarea Java versus implementarea C++.

Evaluati complexitatea-spatiu.

Deadline:

- saptamana 5