**Introducere**

În teoria procesării semnalelor, un *filtru* este un dispozitiv sau un proces care înlătură o parte nedorită dintr-un semnal. Există nenumărate tipuri de filtre, dar cel pe care îl vom studia în această temă este *filtrul median*. Acest tip de filtru este utilizat în mod curent pentru eliminarea zgomotului (*puricilor*) din imagini.

Rezultatul aplicării unui filtru median pe o imagine se poate observa mai jos:

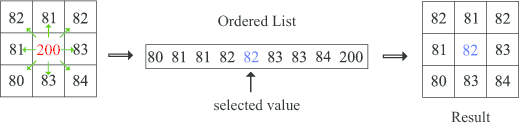


**Reprezentarea unei imagini alb-negru**

În memoria unui calculator, o imagine alb-negru este salvată memorând nivelul de gri pentru fiecare pixel, nivel care este întotdeauna în intervalul [0; 255], unde 0 înseamnă negru și 255 înseamnă alb. Apoi pixelii sunt memorați într-o structură de date arbitrară astfel încât accesul să fie rapid pentru aplicația respectivă. Convenția că pixelul de coordonate (0, 0) este cel din stânga sus, astfel încăt coordonata x crește spre dreapta iar coordonata y crește în jos.

**Aplicarea unui filtru median**

Un filtru median se aplică întotdeauna pe o fereastră de n x n în jurul fiecărui pixel din imaginea originală, unde n este un număr întreg pozitiv impar (3, 5, 7, etc.). Astfel, pentru fiecare pixel din imaginea originală, se extrag toți vecinii săi dintr-o fereastră de n x n în jurul lui. Acești pixeli sunt plasați într-un vector care se sortează. Valoarea pixelului din imaginea filtrată este valoarea din mijlocul vectorului sortat. Pentru o fereastra de 3 x 3:



**Aplicarea filtrului la margini**

La limită, acolo unde fereastra de n x n depășește marginea imaginii, aceasta este bordată prin repetiția pixelilor de la margine. Fie colțul din stânga sus al imaginii originale:

10  20  15 100 ...  
  43  74  75  95 ...  
  45  65  70  80 ...  
  50  55  65  75 ...  
  ... ... ... ...

Dacă se aplică un filtru de 5x5 pentru pixelul din stânga sus (cu valoarea 10), atunci e nevoie de o fereastră pătrată de 5x5 în jurul lui:

10 10  10  20  15

10 10  10  20  15

10 10  **10**  20  15 100 ...

43 43  43  74  75  95 ...  
45 45  45  65  70  80 ...  
   50  55  65  75 ...  
  ... ... ... ...

Cu bold este elementul din colț, iar pentru a obține fereastra de 5x5, s-au replicat elementele de pe margine.

**Fișierele PGM de tip P2**

PGM este un acronim ce vine de la *Portable GrayMap*. Există două formate de fișier PGM - P5, care este un fișier binar, și P2 care este un fișier text. Un fișier în format P2 are următoarea structură:

* Pe prima linie se află codul **P2** care semnalizează tipul fișierului.
* Orice linie care începe cu **#** se consideră comentariu și se ignoră.
* Următoarea linie validă conține două valori (**width** și **height**), ce reprezintă lățimea și înalțimea imaginii
* Următoarea linie validă conține o valoare ce reprezintă valoarea maximă al unui pixel care va fi întâlnit în imagine (deci valoarea corespunzătoare nivelului de alb).
* Pe următoarele linii se află **width x height**valori, separate prin *whitepace*-uri, ce reprezintă nivelurile de gri pentru fiecare pixel, de la stânga la dreapta și apoi de sus în jos.

Exemplu:

P2

# Shows the word "FEEP" (example from Netpbm man page on PGM)

24 7

15

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0

0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0

0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0

0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0

0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



Mai multe despre formatul PGM, [aici](https://en.wikipedia.org/wiki/Netpbm_format" \t "_blank).

**Cerință**

Dându-se un algoritm de sortare dintre BubbleSort și MergeSort, o dimensiune de fereastră și două nume de fișiere cu format PGM-P2, unul de intrare și altul de ieșire, să se aplice un filtru median de dimensiunea specificată pe fiecare pixel din imaginea de intrare, folosind algoritmul specificat pentru sortarea ferestrei, și să se scrie imaginea filtrată în fișierul de ieșire. Ambele fișiere sunt în format PGM-P2.

**Date de intrare**

De la tastatură se dă, pe o singură linie, tipul de algoritm (***bubble*** sau ***merge***), dimensiunea ferestrei, și numele celor două fișiere (de intrare și de ieșire), toate separate prin unul sau mai multe caractere *whitespace*.

Exemplu:

merge 3 test\_in.pgm test\_out.pgm

**Date de ieșire**

Nu se cere afișarea niciunei valori pe stream-ul standard de ieșire. Orice output va fi ignorat de scriptul de evaluare.

**Observații**

1. Scriptul de evaluare va verifica aplicarea corectă a filtrului precum și durata de execuțe a programului, deci aveți grijă ca selecția algoritmului de sortare să aibă efectul scontat (adică să nu sortați cu Merge Sort, indiferent de ce opțiune are utilizatorul).
2. Pentru bordare, în loc să alocați efectiv memorie suplimentară pentru margine (și implicit să aveți matricea cu imaginea mai mare), e mai simplu să scrieți o funcție care să ia ca argument imaginea și coordonatele pixelului și să verificați dacă aceste coordonate sunt în afara limitelor imaginii, situație în care să întoarceți pixelul corespunzător. Atenție în această situație să trimiteți imaginea ca adresă pentru a evita copierea pe stivă a unei cantități foarte mari de date de foarte multe ori. Imaginea din scriptul de notare are rezoluție Full HD: 1920x1080.
3. Puteți folosi o structură de tip **std::vector<std::vector<uint8\_t>>** dar nu este obligatoriu. În această situație, dacă transmiteți acest vector ca argument către o funcție, neapărat acest argument trebuie să fie de tip pointer sau referință (nu s-a discutat la curs conceputul de referință), altfel, imaginea se copiază în întregime la fiecare apel de funcție, și imaginea fiind de rezoluție mare, programul va rula extrem de greu.
4. Testați programul local, pe calculatorul personal înainte de a-l submite. Sunt trei motive:
   1. Este important de distins diferența dintre timpii de procesare pentru ambele tipuri de sortări
   2. Este important de văzut rezultatul filtrării asupra unei imagini sursă
   3. Este mai ușor de depanat programul folosind debugger-ul din CLion
5. Imagine de test: [lena.pgm](https://curs.upb.ro/2023/pluginfile.php/168600/mod_vpl/intro/lena_noise.pgm)

P2

512 512

255