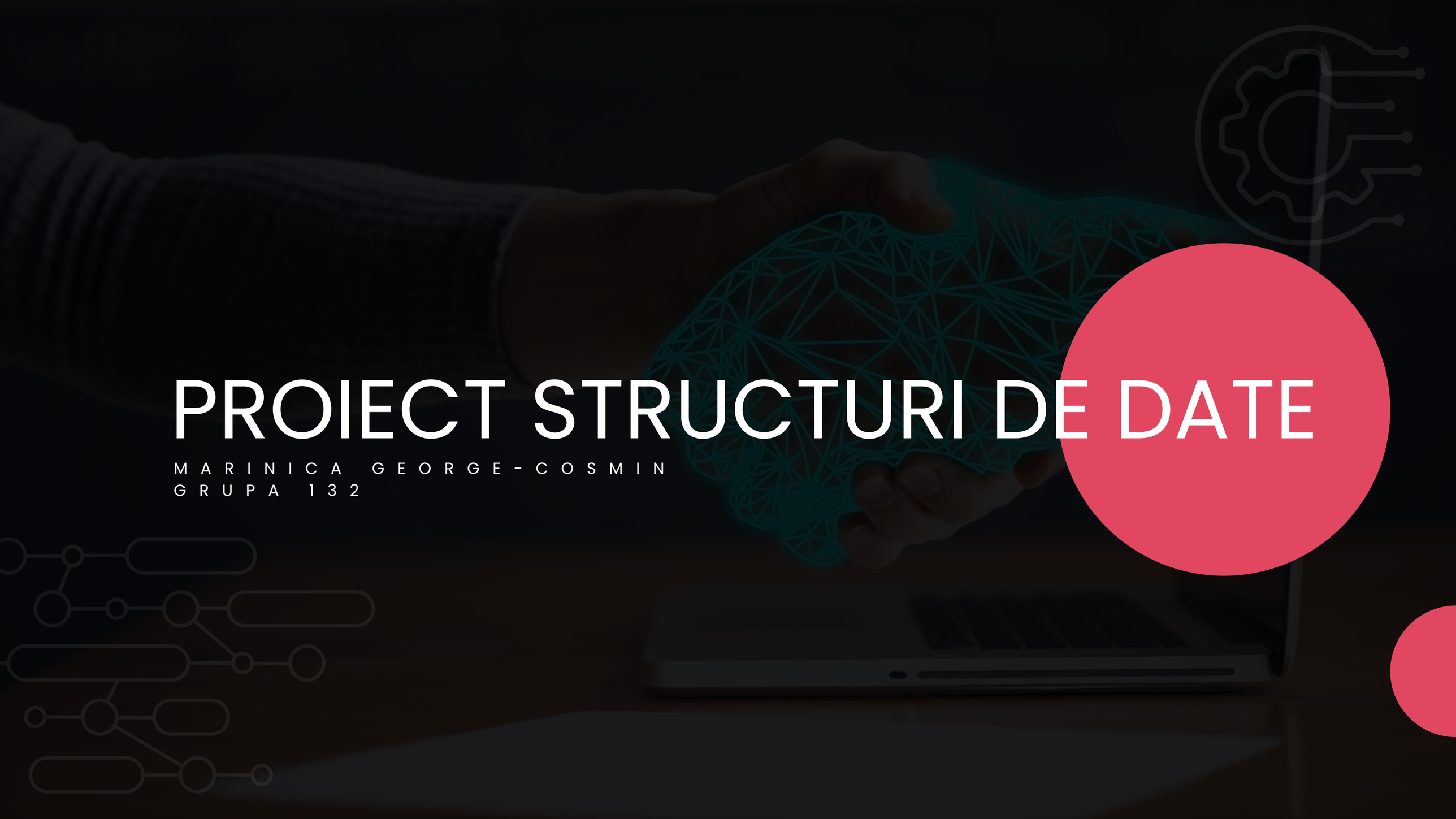


PROIECT STRUCTURI DE DATE

MARINICA GEORGE-COSMIN
GRUPA 132

01

Insert Sort

Eficient pentru seturi mici de numere
Avantajat de simplicitatea programului

02

Merge Sort

Eficienta O(nlogn)
Foloseste metoda divide et impera

03

Quick Sort

Foarte rapid si eficient pentru seturi mici de numere
Eficienta din punct de vedere a spatiului

04

Radix Sort

Foarte eficient seturi mari de numere
Mai rapid decat QuickSort in vectori mari

05

Shell Sort

Foarte rapid pentru seturi de numere aproape sortate
Algoritm usor de implementat - este un InsertSort mai rapid

INSERT SORT

Descriere

Algoritmul este calculat cu numere de la 1 la 1000
Timpul de executie este in milisecunde
Pentru fiecare dimensiune de vector sunt facute 3 teste
Numerele au fost generate cu srand si rand

10^4	10^5	10^6
81ms	7418ms	-
87ms	7393ms	-
80ms	7382ms	-

MERGE SORT

Descriere

Algoritmul este calculat cu numere de la 1 la 1000
Timpul de executie este in milisecunde
Pentru fiecare dimensiune de vector sunt facute 3 teste
Numerele au fost generate cu srand si rand

10^4	10^5	10^6
3ms	44ms	393ms
4ms	51ms	394ms
3ms	46ms	402ms

QUICK SORT

Descriere

Algoritmul este calculat cu numere de la 1 la 1000
Timpul de executie este in milisecunde
Pentru fiecare dimensiune de vector sunt facute 3 teste
Numerele au fost generate cu srand si rand

10^4	10^5	10^6
2ms	49ms	1450ms
2ms	41ms	1374ms
2ms	35ms	1385ms

RADIX SORT

Descriere

Algoritmul este calculat cu numere de la 1 la 1000
Timpul de executie este in milisecunde
Pentru fiecare dimensiune de vector sunt facute 3 teste
Numerele au fost generate cu srand si rand

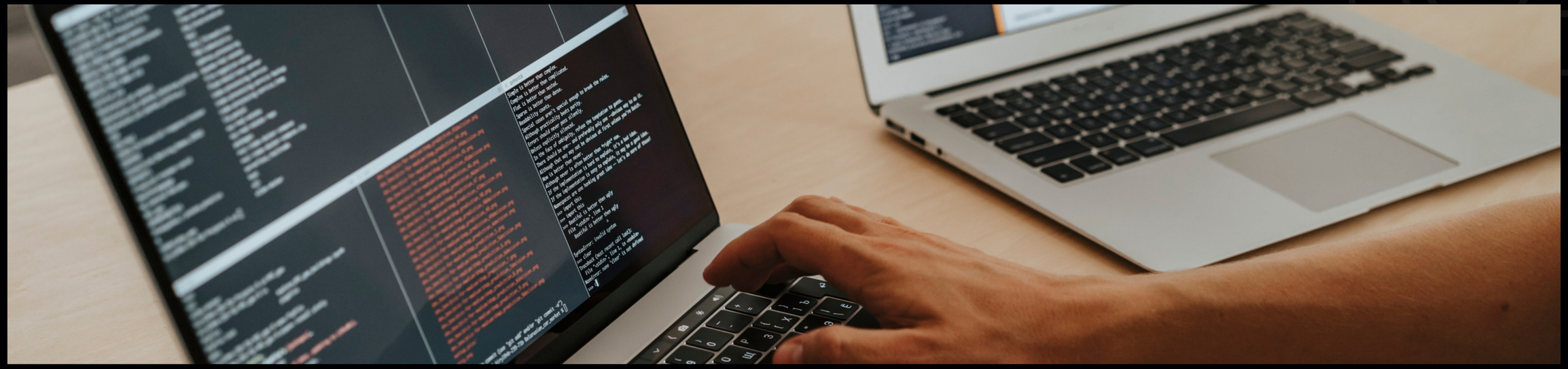
10^4	10^5	10^6
1ms	10ms	-
0ms	11ms	-
0ms	15ms	-

SHELL SORT

Descriere

Algoritmul este calculat cu numere de la 1 la 1000
Timpul de executie este in milisecunde
Pentru fiecare dimensiune de vector sunt facute 3 teste
Numerele au fost generate cu srand si rand

10^3	10^4	10^6
-	-	-
-	-	-
-	-	-



CONCLUZII

Insert Sort este rapid si usor de implementat dar ineficient la vectori mari
Merge Sort este foarte rapid indiferent de lungimea vectorilor
Quick Sort este cel mai rapid pentru vectori cu lungimea sub 10^6
Radix Sort este foarte rapid insa nu l-am putut implica corect pentru a testa vectori cu 10^6 elemente
Nu am putut implementa corect si testa Shell Sort