



SORTARI

Putinelu Andrei Bogdan



TIMPII DE RULARE ALE SORTARILOR

MERGESORT

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0.00201s	0.02699s	0.27099s	3.00003s	32.405s
10^2	0s	0s	0s	0.00200s	0.02809s	0.29325s	3.21219s	34.703s
10^3	0s	0s	0s	0.00300s	0.02899s	0.30120s	3.29710s	35.495s
10^4	0s	0s	0s	0.00307s	0.03099s	0.31620s	3.41299s	36.678s
10^5	0s	0s	0s	0.00300s	0.03199s	0.32600s	3.56166s	38.319s
10^6	0s	0s	0s	0.00310s	0.03000s	0.32955s	3.68171s	39.320s
10^7	0s	0s	0s	0.00317s	0.03099s	0.33623s	3.70210s	40.216s
10^8	0s	0s	0s	0.00315s	0.03199s	0.33399s	3.72815s	41.023s

MERGESORT

- Mergesort-ul se arata ca fiind o metoda destul de buna pentru sortarea numerelor, acesta scotand un timp de sub **0.3 secunde** pentru sortarea a **1.000.000** de numere.
- Pentru **10.000.000** de numere timpul de sortare este aproximativ de **3 secunde**.
- Sortarea deja devine mult mai lenta, scotand un timp de sortare in jur de **40 de secunde** pentru **100.000.000** de elemente.



RADIXSORT (BAZA 2^{16})

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0s	0.00099s	0.01300s	0.12811s	1.30228s
10^2	0s	0s	0s	0s	0.00120s	0.01400s	0.13200s	1.31899s
10^3	0s	0s	0s	0s	0.00199s	0.01398s	0.13099s	1.34503s
10^4	0s	0s	0s	0s	0.00199s	0.01500s	0.15100s	1.70739s
10^5	0s	0s	0s	0s	0.00199s	0.02600s	0.26980s	2.98209s
10^6	0s	0s	0s	0s	0.00299s	0.02799s	0.26899s	3.06421s
10^7	0s	0s	0s	0s	0.00299s	0.02699s	0.27510s	3.09494s
10^8	0s	0s	0s	0s	0.00299s	0.02700s	0.27899s	3.18122s



RADIXSORT (BAZA 2^{16})

- Radixsort reprezinta o metoda buna de sortare, din punct de vedere al timpului de sortare, putand sorta pana la **10.000.000** de numere sub **0.3 secunde**.
- In cazul sortarii a **100.000.000** numere, radixsort scoate un timp impresionant din intervalul **1 secunda – 3 secunde**, depinzand de valoarea numerelor (in acest caz valoarea maxima pe care o pot avea numerele pentru a sorta in **1 secunda** fiind **10.000**, iar pentru **3 secunde 100.000.000**).

RADIXSORT (BAZA 10)

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0s	0.00099s	0.01500s	0.15316s	1.44619s
10^2	0s	0s	0s	0s	0.00300s	0.02800s	0.26409s	2.59448s
10^3	0s	0s	0s	0s	0.00400s	0.03900s	0.38600s	3.81504s
10^4	0s	0s	0s	0s	0.00599s	0.51466s	0.51466s	4.97681s
10^5	0s	0s	0s	0s	0.00599s	0.06400s	0.62220s	6.14699s
10^6	0s	0s	0s	0s	0.00799s	0.07500s	0.76151s	7.34878s
10^7	0s	0s	0s	0s	0.00899s	0.08999s	0.86157s	8.59516s
10^8	0s	0s	0s	0s	0.01000s	0.09800s	0.96102s	9.58609s



RADIXSORT (BAZA 10)

- Radixsort in baza 10, de asemenea, scoate un timp de sortare bun, sortand pana la **1.000.000** numere sub **0.1 secunde**.
- Pentru **10.000.000** numere timpul incepe sa creasca si sa fluctueze in functie de valorile pe care le pot lua numerele ce vor fi sortate – timpul crescand de la **0.15 secunde** pentru valoarea maxima a numerelor: **10**, la **1 secunda** pentru valoarea maxima a numerelor: **100.000.000**.
- Aceeasi poveste este valabila si pentru sortarea a **100.000.000** numere: pentru numere cu valoarea maxima **10** timpul de sortare este **1.5 secunde**, iar pentru numere cu valoarea maxim **100.000.000** timpul de sortare este **9.6 secunde**.

Radixsort (baza 2^{16})

- Timpii sunt identici pentru sortarea a pana la **10.000** numere (in cazurile cu valori maxime de la **10** la **100.000.000**)
- Diferenta apare de la sortarea a **100.000** numere pana la **100.000.000** numere (in cazurile cu valori maxime de la **10** la **100.000.000**), radixsort in baza 2^{16} avand timpii de, intre **2-5** ori, mai mici decat radixsort in baza **10**.



Radixsort (baza 10)

- Diferenta nu este, insa, una imensa in toate cazurile, deoarece majoritatea cazurilor sorteaza sub **1 secunda**.
- In cazurile in care sortarea se face in mai multe secunde intervine diferenta notabila (de exemplu radixsort (**10**) sorteaza **100.000.000** numere cu valori pana la **100.000.000** in **10 secunde**, pe cand radixsort(2^{16}) sorteaza in **3 secunde**).

SHELLSORT

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0.00100s	0.01100s	0.14099s	1.61551s	18.556s
10^2	0s	0s	0s	0.00201s	0.01800s	0.20599s	2.26224s	27.028s
10^3	0s	0s	0s	0.00200s	0.02399s	0.30500s	3.47109s	38.851s
10^4	0s	0s	0s	0.00200s	0.02799s	0.34918s	4.82862s	59.456s
10^5	0s	0s	0s	0.00200s	0.03011s	0.41999s	5.68828s	84.801s
10^6	0s	0s	0s	0.00199s	0.02899s	0.41699s	6.21055s	90.861s
10^7	0s	0s	0s	0.00201s	0.02999s	0.43798s	6.09241s	89.855s
10^8	0s	0s	0s	0.00200s	0.02900s	0.42741s	6.09723s	92.375s

SHELLSORT

- Timpul de sortare al shellsort-ului este de sub **0.5 secunde** pentru tablouri cu pana la **1.000.000** de numere.
- Pentru **10.000.000** de numere timpul este decent, variind intre **1.6 secunde** pentru elemente mai mici ca **10** si **6 secunde** pentru elemente mai mici ca **100.000.000**.
- Deja la **100.000.000** de elemente timpul de sortare se mareste considerabil, pentru elemente mai mici ca **10** este **18.6 secunde**, iar pentru elemente mai mici ca **100.000.000** este de **92.5 secunde**.



SHELLSORT KNUTH

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0.00100s	0.00899s	0.09600s	1.08100s	12.061s
10^2	0s	0s	0s	0.00099s	0.01499s	0.16400s	1.71941s	19.234s
10^3	0s	0s	0s	0.00200s	0.02099s	0.25199s	3.15562s	33.204s
10^4	0s	0s	0s	0.00200s	0.02400s	0.33326s	4.57960s	56.662s
10^5	0s	0s	0s	0.00100s	0.02600s	0.36599s	5.25790s	81.643s
10^6	0s	0s	0s	0.00189s	0.02699s	0.37800s	5.82372s	90.388s
10^7	0s	0s	0s	0.00201s	0.02499s	0.37264s	5.87666s	89.679s
10^8	0s	0s	0s	0.00200s	0.02700s	0.40101s	5.95665s	91.216s

SHELLSORT KNUTH

- Folosind secventa lui Knuth pentru shellsort, am obtinut niste timpi de sortare rapizi, de sub **0.5 secunde** pentru tablouri cu pana la **1.000.000** de numere.
- Pentru **10.000.000** de numere timpul este decent, fiind cuprins intre **1 secunda** pentru elemente mai mici ca **10** si **6 secunde** pentru elemente mai mici ca **100.000.000**.
- Deja la **100.000.000** de elemente timpul de sortare se mareste considerabil, pentru elemente mai mici ca **10** este **12 secunde**, iar pentru elemente mai mici ca **100.000.000** este de **91.2 secunde**.



Shellsort

- Timpii sunt aproximativ egali pentru sortarea a pana la **1.000.000** numere (in cazurile cu valori maxime de la **10** la **100.000.000**) .
- Diferenta vizibila incepe sa apara de la sortarea a **10.000.000** numere (in cazurile cu valori maxime de la **10** la **100.000.000**), shellsort knuth avand un timp mai rapid decat shellsort cu o mica diferenta de intre **0.1 secunde - 0.6 secunde**.



Shellsort Knuth

- In cazul sortarii a **100.000.000** elemente (in cazurile cu valori maxime de la **10** la **100.000.000**) shellsort knuth are un timp de sortare mai rapid ca shellsort, diferenta masurandu-se in secunde (intre **1 secunda – 8 secunde**).

QUICKSORT (CU PIVOT ULTIMUL ELEMENT)

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0.01200s	1.10938s	-	-	-
10^2	0s	0s	0s	0.00200s	0.13211s	-	-	-
10^3	0s	0s	0s	0.00100s	0.02499s	1.42069s	-	-
10^4	0s	0s	0s	0.00101s	0.01599s	0.27022s	14.7075s	-
10^5	0s	0s	0s	0.00100s	0.01599s	0.18299s	3.02119s	149.712s
10^6	0s	0s	0s	0.00200s	0.01600s	0.18299s	2.16918s	33.439s
10^7	0s	0s	0s	0.00101s	0.01700s	0.19000s	2.18434s	24.196s
10^8	0s	0s	0s	0.00099s	0.01801s	0.19300s	2.17543s	24.4606s



QUICKSORT (CU PIVOT ULTIMUL ELEMENT)

- Quicksort cu pivot ultimul element functioneaza mult mai rapid pe elemente cat mai diferite.
- Pentru elemente cu valoarea maxima **10** programul cedeaza cand are de sortat mai mult de **100.000** numere.
- Diferenta data de valoarea numerelor ce vor fi sortate este notabila, timpul de sortare trecand de la **150 de secunde** pentru **100.000.000** de numere cu valoarea maxima **100.000** la **33 de secunde** pentru **100.000.000** de numere cu valoarea maxima **1.000.000**.

QUICKSORT CU MEDIANA DIN 3

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0s	0.00882s	0.10399s	1.16499s	13.268s
10^2	0s	0s	0s	0.00099s	0.01000s	0.11299s	1.24173s	13.879s
10^3	0s	0s	0s	0.00099s	0.01099s	0.12099s	1.33100s	14.939s
10^4	0s	0s	0.00099s	0.00100s	0.01299s	0.13000s	1.40121s	15.473s
10^5	0s	0s	0.00100s	0.00100s	0.01499s	0.14333s	1.52268s	16.298s
10^6	0s	0s	0s	0.00100s	0.01300s	0.15300s	1.65052s	17.472s
10^7	0s	0s	0s	0.00100s	0.01400s	0.14900s	1.68200s	18.566s
10^8	0s	0s	0s	0.00100s	0.01400s	0.14800s	1.69676s	19.039s



QUICKSORT CU MEDIANA DIN 3

- Quicksort cu mediana din 3 are niste timpi foarte buni de sortare, indiferent de valoarea numerelor sortate.
- Pentru pana la **1.000.000** numere, timpii de sortare sunt apropiati pentru numerele cu valoarea maxima **10** si cele cu valoare maxima **100.000.000**, fiind sortate in sub **0.15 secunde**.
- Pentru **10.000.000** de numere timpul de sortare este in jur de **1.5 secunde**, iar pentru **100.000.000** de numere oscileaza intre **13-19 secunde**.

Quicksort (cu pivot ultimul element)

- Diferenta intre cele 2 variante de quicksort este semnificativa, un prim aspect fiind faptul ca quicksort cu pivot ultimul element nu poate sorta in anumite cazuri de combinatii dintre valoare maxima si numar elemente, in timp ce quicksort cu mediana din 3 nu prezinta aceasta problema.



Quicksort cu mediana din 3

- In cazurile in care poate sorta, quicksort (mediana 3) are timpii de sortare mult mai buni decat quicksort (last), de exemplu **16 secunde** versus **150 de secunde**.
- Pentru anumite combinatii dintre valoare maxima si numar elemente quicksort (last) are timpi relativ similari cu quicksort (mediana 3).

COUNTSORT

Valoare maxima Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0s	0.00099s	0.00800s	0.08099s	0.78823s
10^2	0s	0s	0s	0s	0.00099s	0.00800s	0.07899s	0.80700s
10^3	0s	0s	0s	0s	0.00099s	0.00900s	0.08100s	0.79056s
10^4	0s	0s	0s	0s	0.00099s	0.00899s	0.08299s	0.80221s
10^5	0.00199s	0.00200s	0.00101s	0.00200s	0.00299s	0.00999s	0.08600s	0.83158s
10^6	0.01200s	0.01110s	0.01201s	0.01301s	0.01500s	0.03100s	0.11299s	0.94215s
10^7	0.09100s	0.11900s	0.11900s	0.11900s	0.12799s	0.15700s	0.42308s	2.36225s
10^8	1.09148s	1.13312s	1.11618s	1.12342s	1.14120s	1.24501s	1.59937s	4.5243s

COUNTSORT

- Countsort are cei mai buni timpi de sortare de pana acum, sortand sub **1 secunda** pana la **100.000.000** numere cu valoarea maxima de la **10** la **1.000.000** si pana la **10.000.000** numere cu valoarea maxima **10.000.000**.
- In rest pana la **10.000.000** numere cu valoarea maxima **100.000.000** sunt sortate in jur de **1 secunda**, singurele cazuri de sortare cu mai mult de **1 secunda** fiind:
 - **100.000.000** numere cu valoare **10.000.000** in **2.3 secunde**
 - **100.000.000** numere cu valoare **100.000.000** in **4.5 secunde**



NATIVESORT

Valoare maxima \ Numar de elemente	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8
10^1	0s	0s	0s	0.00099s	0.01299s	0.15017s	1.74051s	20.106s
10^2	0s	0s	0s	0.00100s	0.01400s	0.15599s	1.81535s	20.992s
10^3	0s	0s	0s	0.00199s	0.01500s	0.17000s	1.94344s	21.913s
10^4	0s	0s	0s	0.00199s	0.01699s	0.17700s	2.01366s	22.741s
10^5	0s	0s	0s	0.00207s	0.01900s	0.19399s	2.06387s	23.261s
10^6	0s	0s	0s	0.00203s	0.01800s	0.21099s	2.27463s	24.145s
10^7	0s	0s	0s	0.00199s	0.01899s	0.22124s	2.39116s	26.031s
10^8	0s	0s	0s	0.00107s	0.01800s	0.20799s	2.38015s	27.140s



NATIVESORT

- Nativesort sorteaza pana la **1.000.000** numere cu valori de la **10** la **100.000.000** sub **0.2 secunde**.
- Timpul ramane decent la sortarea a **10.000.000** numere cu valori de la **10** la **100.000.000**, acesta fiind in jur de **2 secunde**.
- Insa la sortarea a **100.000.000** numere cu valori de la **10** la **100.000.000** timpul sare la un interval mult mai mare, de **20-27 de secunde**.

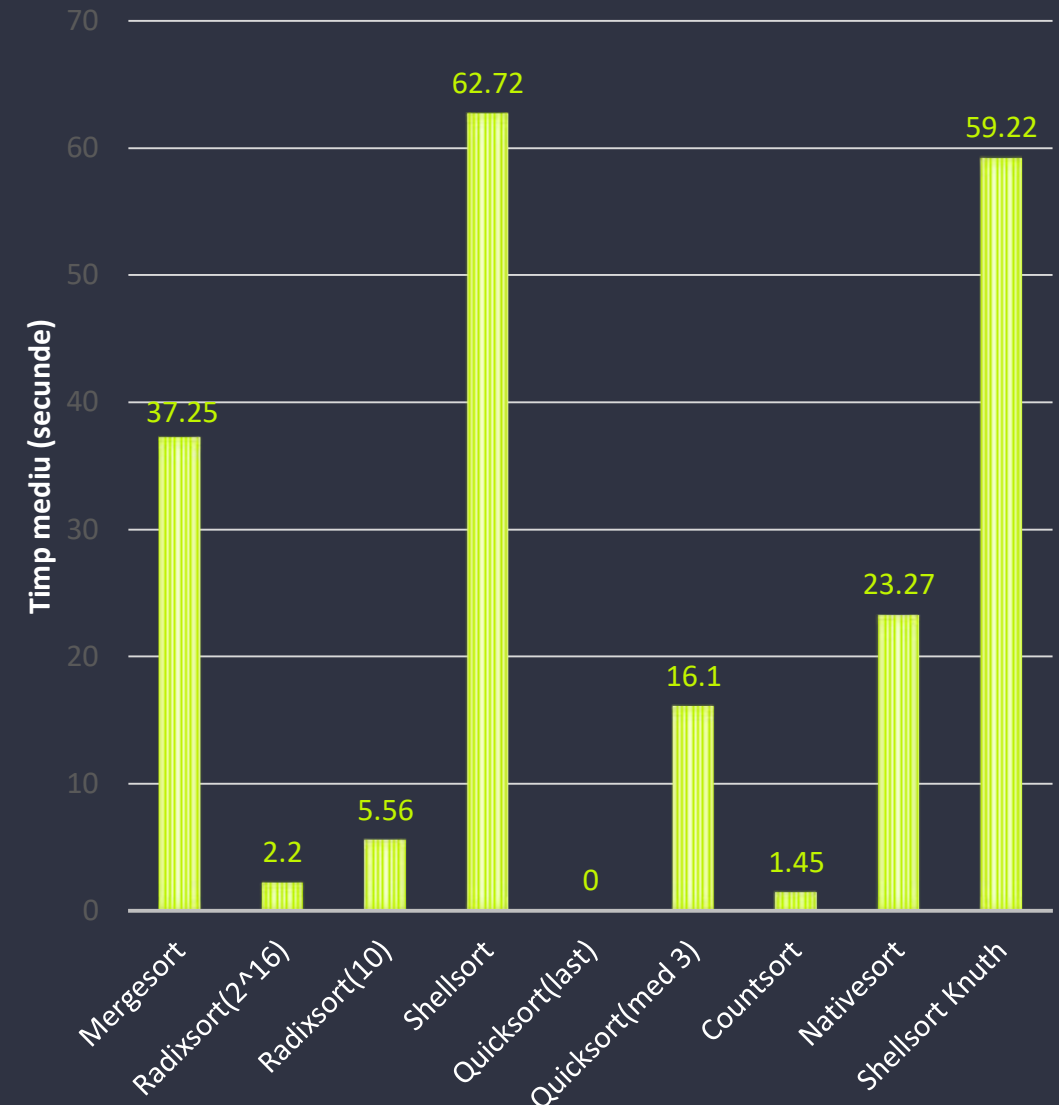
The background is a high-angle, black and white photograph of a dense urban landscape, showing numerous skyscrapers and buildings. Overlaid on this is a graphic consisting of a bright green rectangular area on the left, a white L-shaped frame, and a dark gray rectangular area on the right where the text is located.

ANALIZA ALGORITMIOR

ANALIZA TIMPULUI MEDIU

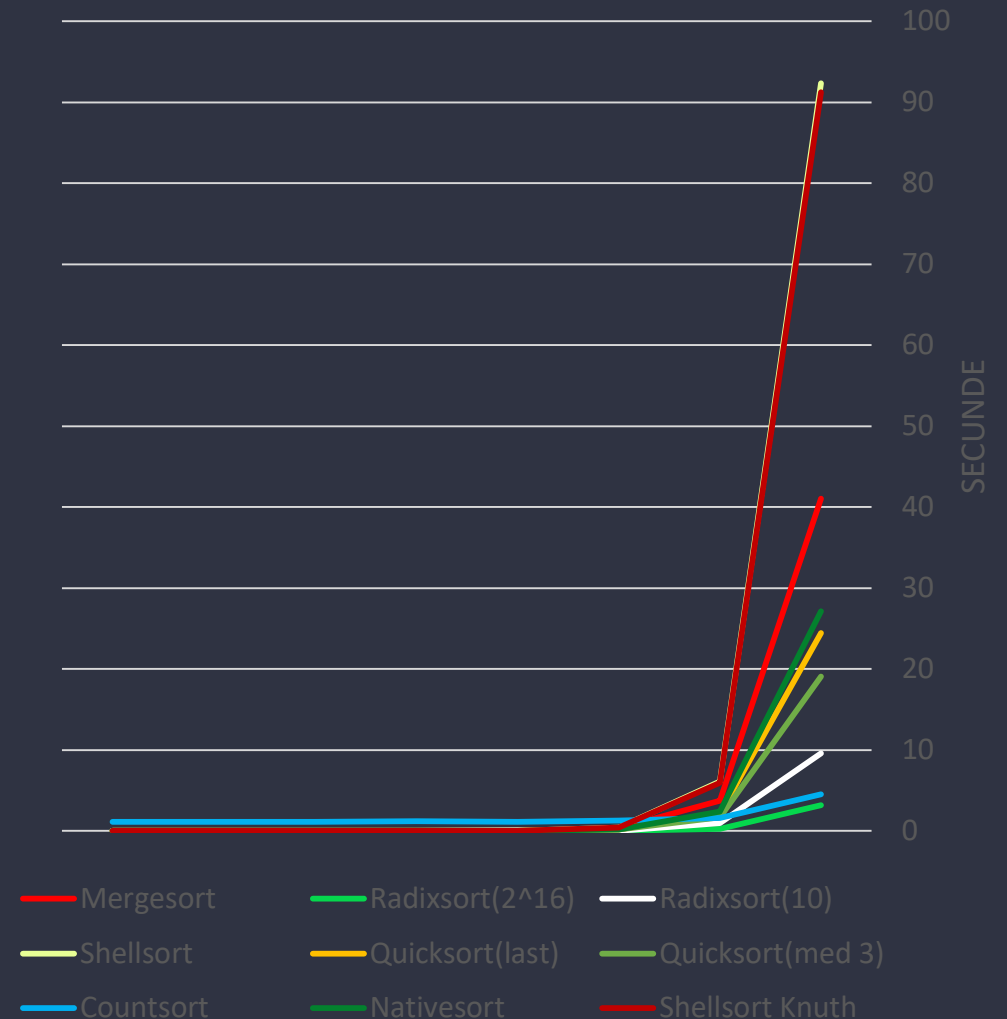
- Graficul a fost creat luand in calcul timpii de sortare a **100.000.000** elemente de catre algoritmi de sortare, sortand pe rand elemente cu valoare maxima **10, 100, ..., 100.000.000**.
- Quicksort(last) a fost descalificat, deoarece nu poate sorta atatea elemente, decat daca pot avea valoarea maxima de la **10.000** in sus.
- Se pot observa diferentele dintre algoritmi, countsort-ul avand cea mai buna medie a timpului de sortare.
- Se poate observa si diferenta dintre cele 2 variante de radixsort, cea in baza 2^{16} , fiind mai rapida ca cea in baza 10.
- De asemenea, quicksort(media 3) surclaseaza quicksort(last), in primul rand pentru ca poate sorta toate testele propuse, iar apoi datorita timpului mediu (a fost o instanta cand quicksort(last) a sortat un test in **146 de secunde**).
- Cele 2 variante de shellsort sunt cele mai lente, aflandu-se la o diferenta notabila fata de celelalte sortari.

100.000.000 ELEMENTE



ANALIZA PERFORMANTA

- Graficul analizeaza evolutia tuturor algoritmilor de la sortarea a 10 elemente la sortarea a **100.000.000** elemente (cu valoarea maxima **100.000.000**).
- Dupa cum se observa algoritmii cu rezultatele cele mai bune sunt radixsort (2^{16}) si countsort.
- Nativesort are o evolutie destul de buna, ultimul test facand diferenta la timpul de sortare.



CONCLUZII

- Potrivit timpilor de sortare si comparatiilor dintre acestia, countsort si radixsort(2^{16}) sunt sortarile cu cele mai bune performante.
- Cand vine vorba de sortarea unui numar considerabil de elemente cu valori mari, este importanta alegerea unui algoritm rapid.
- In cazul in care avem mai putine elemente, algoritmi prezinta un timp similar de sortare, fiind folosite utilizarea lor.



THANK YOU