



# Sortări

RadixSort  
MergeSort  
ShellSort  
CountingSort  
HeapSort

Proiect realizat de  
Asăvoaei Bianca Gabriela

# Conținutul prezentării

- A. Pentru fiecare din cele 5 tipuri de sortări se va specifica:
  - 1.Complexitate
  - 2.Complexitate spațiu
  - 3.Stabilitate
  - 4.Daca se bazează pe comparații sau nu
  
- B. Comparații între timpii de rulare pe mai multe teste cu numere naturale

Sortare	RadixSort	MergeSort	ShellSort	CountingSort	HeapSort
Complexitate	$O(n+k)$	$O(n \log n)$	$O(n \log n)$	$O(n+k)$	$O(n \log n)$
Complexitate spațiu	$O(\max)$	$O(n)$	$O(1)$	$O(\max)$	$O(1)$
Stabilitate	Da	Da	Nu	Da	Nu
Criteriul comparație	Nu	Da	Nu	Nu	Da

	$N = 10^2$ $V_{\max} = 10^3$	$N = 10^3$ $V_{\max} = 10^4$	$N = 10^4$ $V_{\max} = 10^4$	$N = 10^5$ $V_{\max} = 10^5$
RadixSort	<b>0.008 milisecunde</b>	<b>0.01 milisecunde</b>	<b>0.048 milisecunde</b>	<b>1.235 milisecunde</b>
ShellSort	0.056 milisecunde	0.193 milisecunde	0.661 milisecunde	15.162 milisecunde
MergeSort	<b>0.262 milisecunde</b>	<b>1.368 milisecunde</b>	<b>9.394 milisecunde</b>	<b>146.459 milisecunde</b>
HeapSort	<b>0.042 milisecunde</b>	0.526 milisecunde	0.196 milisecunde	2.632 milisecunde
CountingSort	0.022 milisecunde	0.067 milisecunde	<b>0.047 milisecunde</b>	<b>1.007 milisecunde</b>
STL	0 milisecunde	0.01 milisecunde	0 milisecunde	0 milisecunde

	$N = 10^6$ $V_{\max} = 10^6$	$N = 10^7$ $V_{\max} = 10^5$	$N = 10^8$ $V_{\max} = 10^2$	$N = 10^8$ $V_{\max} = 10^8$
RadixSort	<b>3.536</b> <b>milisecunde</b>	<b>45.521</b> <b>milisecunde</b>	<b>493.323</b> <b>milisecunde</b>	<b>350.188</b> <b>milisecunde</b>
ShellSort	128.877 milisecunde	2138.65 milisecunde	22360.5 milisecunde	13371.7 milisecunde
MergeSort	<b>1014.16</b> <b>milisecunde</b>	<b>13112.8</b> <b>milisecunde</b>	<b>131450</b> <b>milisecunde</b>	<b>88492.8</b> <b>milisecunde</b>
HeapSort	26.881 milisecunde	792.143 milisecunde	2862.78 milisecunde	2750.79 milisecunde
CountingSort	7.117 milisecunde	<b>1.552</b> <b>milisecunde</b>	<b>0.014</b> <b>milisecunde</b>	805.974 milisecunde
STL	0 milisecunde	0 milisecunde	0 milisecunde	0 milisecunde

# Observații



Radix Sort și Counting Sort sunt cei mai rapizi algoritmi de sortare dintre cele cinci. Marele dezavantaj al Counting Sort-ului este că folosește multă memorie. Avantajul Radix Sort-ului este că poate sorta numere oricât de mari, fiind mai practic decât Counting.



Merge Sort este cel mai lent algoritm de sortare dintre cele cinci. Este un algoritm care se bazează pe Divide et Impera și are space complexity de  $O(n)$ , ceea ce înseamnă că ocupă mult spațiu și încetinește operațiile.

<b><math>N = 10^6</math> <math>V_{\max} = 10^6</math></b>	<b>Baza = 2</b>	<b>Baza = 10</b>	<b>Baza = 256</b>	<b>Baza = 65536</b>
<b>RadixSort</b>	<b>3.619 milisecunde</b>	<b>3.06 milisecunde</b>	<b>2.886 milisecunde</b>	<b>2.735 milisecunde</b>



Observăm că pe măsură ce creștem baza, scade timpul de execuție, însă diferența dintre ultimele 2 baze nu este mare.