Nume și prenume : Ion Andrei

Grupa: 324CC

Tema 1 – Analiza algoritmilor.

Vector sortat crescător

Sortare	10 elemente	300 elemente	10000 elemente
Cocktail Sort	26	896	29 996
Block Sort	211	41 790	2 658 078
Binary Tree Sort	415	316 950	350 069 821
Heap Sort	1 248	92 443	4 942 364
Radix Sort	2 192	42 883	1 400 883
Splay Tree Sort	413	13 463	454 250
Strand Sort	874	396 144	-

Vector generat aleator

Sortare	10 elemente	300 elemente	10000 elemente
Cocktail Sort	318	431 786	501 760 405
Block Sort	315	52 950	3 033 174
Binary Tree Sort	268	18 330	1 068 090
Heap Sort	1 209	85 700	4 678 026
Radix Sort	2 192	42 883	1 400 883
Splay Tree Sort	785	82 644	4 908 332
Strand Sort	1 114	460 442	-

Vector sortat descrescător

Sortare	10 elemente	300 elemente	10000 elemente
Cocktail Sort	905	897 150	999 882 502
Block Sort	571	63 102	3 293 678
Binary Tree Sort	325	227 250	215 827 295
Heap Sort	919	77 190	4 406 764
Radix Sort	2 192	42 883	1 400 883
Splay Tree Sort	395	12 865	502 627
Strand Sort	1149	556 789	-

Comparaţie între algoritmi.

Vectori sortaţi crescător:

- Algoritmii Cocktail Sort si Splay Tree Sort sunt foarte eficienti pe valori deja sortate. Cocktail Sort ruleaza in O(n) iar Splay Tree Ruleaza intr-un O(k * lgn *n)
- Binary Search Tree-ul, cand vectorul este sortat, este foarte ne-eficient deoarece elementele trebuie sa respecte conditia arborelui BST.

Vectori generaţi aleator:

- Pe vectori generati aleatoriu, cu valori mari, Radix Sort si Binary Tree Sort au o complexitate foarte buna, O(n *lgn)<= T(n)<= O(n * lgn * lgn) comparativ cu Cocktail Sort care este in O(k*n^2)
- Radix sort, pe orice tip de vector, daca inserarea o constanta bine definita atunci complexitatea nu variaza! Ramane in O(n).

Vector sortaţi descrescător:

- Pe vectori sortati descrescator cele mi eficiente sortari sunt Binary Tree Sort si Splay Tree Sort.
- Cel mai slab algoritm de sortare, dpdv al timpului, este Cocktail sort, ajungand la (k*n^2) iar k destul de mare...
- Cel mai eficient este SplayTree sort, ajungand, pe 10 000 de elemente, la O(n*lgn)

Observatie:

- Heap sort-ul si Radix sort-ul sunt cei mai stabili algoritmi din punct de vedere al complexitatii, pe orice tip de vectori.

Concluzie personala:

- Algoritmul meu preferat este Radix Sort, deoarecere isi pastreaza complexitatea pe orice tip de vector atata timp cand se presupune inserarea constanta.
- Cel mai dificil de implmentat algoritm a fost Strand Sort-ul. Acest algoritm, pe langa faptul ca mi s-a parut dificil nici nu prea are importanta mare in practica. Ajunge la O(n*lgn), ca aproape oricare alt algoritm din aceasta tema. In plus, implementarea mea ajunge in O(n^3), poate chiar O(n^4) (dar nu asimptotic).