Structuri de date - tema1

Burdusa Petru-Robert

Pentru aceasta tema am implementat 5 algoritmi de sortare si i-am comparat atat intre ei cat si cu algoritmul nativ de sortare din c++ folosind 10 teste generate random.

Codul pentru algoritmii de sortare se gaseste in "sortari.cpp" iar cel pentru generatorul de numere in "generator.cpp".

Algoritmii implementati au fost bubble sort, count sort, merge sort, radix sort(in baza 16 si baza 2^10) si quick sort(cu pivot ales random). Mai jos se pot observa timpii de executie ai fiecarui algoritm pe fiecare test(N = numarul de numere, MAX = numarul maxim pana la care pot ajunge numerele generate).

TEST 1

N = 10000 MAX = 10000

BUBBLE SORT TIME: 750.329ms SORTED

COUNT SORT TIME: 0ms SORTED

RADIX SORT BASE 1024 TIME: 0ms SORTED

RADIX SORT BASE 16 TIME: 0ms SORTED

MERGE SORT TIME: 12.21ms SORTED

QUICK SORT RANDOM TIME: 20.4384ms SORTED

C++ NATIVE SORT RANDOM TIME: 0ms SORTED

TEST 2

N = 10000 MAX = 10000000

BUBBLE SORT TIME: 746.242ms SORTED

COUNT SORT TIME: 38.6977ms SORTED

RADIX SORT BASE 1024 TIME: 0ms SORTED

RADIX SORT BASE 16 TIME: 8.0803ms SORTED

MERGE SORT TIME: 10.0526ms SORTED

QUICK SORT RANDOM TIME: 15.625ms SORTED

C++ NATIVE SORT RANDOM TIME: 0ms SORTED

TEST 3

N = 10000000 MAX = 10000

BUBBLE SORT TIME: -

COUNT SORT TIME: 389.676ms SORTED

RADIX SORT BASE 1024 TIME: 1260.38ms SORTED

RADIX SORT BASE 16 TIME: 2139.1ms SORTED

MERGE SORT TIME: 17813.5ms SORTED

QUICK SORT RANDOM TIME: 7022.29ms SORTED

C++ NATIVE SORT RANDOM TIME: 3455.58ms SORTED

TEST 4

N = 50000 MAX = 50000

BUBBLE SORT TIME: 18233.6ms SORTED

COUNT SORT TIME: 8.0688ms SORTED

RADIX SORT BASE 1024 TIME: 8.0189ms SORTED

RADIX SORT BASE 16 TIME: 10.0617ms SORTED

MERGE SORT TIME: 80.4341ms SORTED

QUICK SORT RANDOM TIME: 90.6478ms SORTED

C++ NATIVE SORT RANDOM TIME: 10.0641ms SORTED

TEST 5

N = 10000000 MAX = 10000000

BUBBLE SORT TIME: -

COUNT SORT TIME: 802.72ms SORTED

RADIX SORT BASE 1024 TIME: 1773.03ms SORTED

RADIX SORT BASE 16 TIME: 3078.28ms SORTED

MERGE SORT TIME: 18388.9ms SORTED

QUICK SORT RANDOM TIME: 20934.4ms SORTED

C++ NATIVE SORT RANDOM TIME: 3970.19ms SORTED

TEST 6

N = 100000000 MAX = 10000000

BUBBLE SORT TIME: -

COUNT SORT TIME: 6984.52ms SORTED

RADIX SORT BASE 1024 TIME: 17442.7ms SORTED

RADIX SORT BASE 16 TIME: 30427.1ms SORTED

MERGE SORT TIME: -

QUICK SORT RANDOM TIME: -

C++ NATIVE SORT RANDOM TIME: 45671.5ms SORTED

TEST 7

N = 10000 MAX = 1000000000

BUBBLE SORT TIME: 747.356ms SORTED

COUNT SORT TIME: -

RADIX SORT BASE 1024 TIME: 0ms SORTED

RADIX SORT BASE 16 TIME: 2.0583ms SORTED

MERGE SORT TIME: 18.1229ms SORTED

QUICK SORT RANDOM TIME: 20.619ms SORTED

C++ NATIVE SORT RANDOM TIME: 0.9302ms SORTED

TEST 8

N = 5000000 MAX = 2000000000

BUBBLE SORT TIME: -

COUNT SORT TIME: -

RADIX SORT BASE 1024 TIME: 1124.05ms SORTED

RADIX SORT BASE 16 TIME: 1996.43ms SORTED

MERGE SORT TIME: 8936.46ms SORTED

QUICK SORT RANDOM TIME: 14461.9ms SORTED

C++ NATIVE SORT RANDOM TIME: 2001.93ms SORTED

TEST 9

N = 5000000 MAX = 100

BUBBLE SORT TIME: -

COUNT SORT TIME: 192.311ms SORTED

RADIX SORT BASE 1024 TIME: 328.725ms SORTED

RADIX SORT BASE 16 TIME: 573.722ms SORTED

MERGE SORT TIME: 8565.69ms SORTED

QUICK SORT RANDOM TIME: 1435.89ms SORTED

C++ NATIVE SORT RANDOM TIME: 1626.86ms SORTED

TEST 10

N = 100000000 MAX = 2000000000

BUBBLE SORT TIME: -

COUNT SORT TIME: -

RADIX SORT BASE 1024 TIME: 22563.7ms SORTED

RADIX SORT BASE 16 TIME: 39772.8ms SORTED

MERGE SORT TIME: -

QUICK SORT RANDOM TIME: -

C++ NATIVE SORT RANDOM TIME: 47026.4ms SORTED

Concluzii:

-dupa cum era de asteptat, bubble sort este cel mai slab algoritm de sortare, fiind singurul ce atinge complexitatea O(N^2), din acest motiv nu l-am lasat sa ruleze pe teste cu mai mult de 50 000 de numere

-count sort este cel mai rapid pe testele cu multe numere, dar consuma multa memorie(O(MAX)), si din acest motiv nu poate rula pe testele cu numere mai mari de 10 000 000

-merge sort si quick sort sunt urmatoarele ca timp dupa bubble sort, dar la o diferenta considerabila de el, avand totusi complexitatea O(n log n)

-radix sort este mai rapid decat quick sort si merge sort, iar cel in baza 2^10 este clar mai rapid decat cel in baza 16. Este totusi mai lent decat count sort dar are avantajul ca merge pe numere oricat de mari.

-Algoritmul de sortare nativ din c++ mai rapid decat quick si merge sort, dar ceva mai lent decat radix, fiind limitat de faptul ca sorteaza prin comparatii.