Alexandru Andrei

Grupa 132

**PROIECT Structuri de Date**

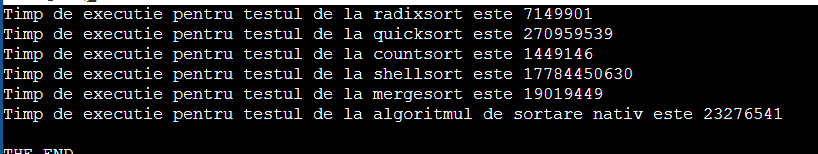
Mai jos voi prezenta timpii de rulare pentru fiecare algoritm in parte.  
Pentru verificarea corectitudinii fiecarui algoritm am facut o functie bool care returneaza 1 daca metoda de sortare intoarce vectorul sortat corect.

***TIMPII PREZENTATI MAI JOS SUNT MASURATI IN NANOSECUNDE!***

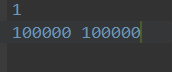
Primul test:



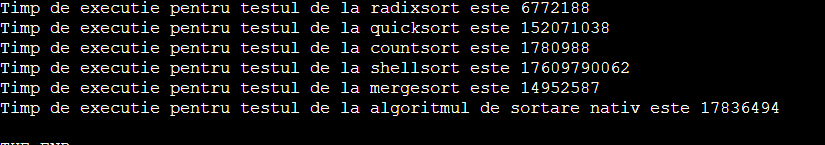
Timpii de executie pentru primul test:



Al doilea test:



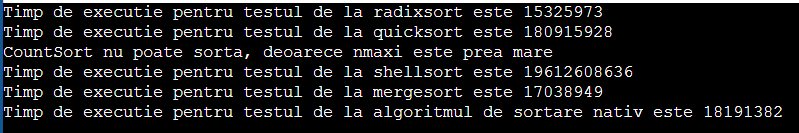
Timpii de executie pentru al doilea test:



Al treilea test:



Timpii de executie pentru al treilea test:



Asadar, avem urmatoarele concluzii:

MergeSort este rapid pentru input-uri cu multe numere, care sunt partial sortate intre ele.   
 este incet pentru input-uri cu putine numere, numere sortate descrescator.

QuickSort este rapid pentru seturi cu multe numere care sunt random.

este incet pentru input-uri in care numerele sunt deja sortate/cu multe valori egale.

CountSort este rapid pentru input-uri in care diferenta dintre valoarea maxima si minima e relative mica.

este incet pentru input-uri in care sunt multe valori egale intre ele sau diferenta dintre minim si maxim este mare.

ShellSort este rapid pentru input-uri aproape sortate deja.

este incet pentru input-uri in care sunt multe inversiuni.

RadixSort este rapid pentru input-uri cu multe numere care au acelasi numar de cifre.

este incet pentru input-uri in care sunt putine numere, diferenta dintre numarul de cifre fiind mare.