Лабораторна робота №5

Спеціальні алгоритми сортування

Обрав сортування Bucket sort (блочне сортування)і Radix sort (порозрядне сортування).

Псевдокод:

блочне сортування:

function bucket-sort(A, n) is

buckets ← новый массив из n пустых элементов

for i = 0 to (length(A)-1) do

вставить A[i] в конец массива buckets[msbits(A[i], k)]

for i = 0 to n - 1 do

next-sort(buckets[i])

return Конкатенация массивов buckets[0], ..., buckets[n-1]

порозрядне сортування:

Ідея полягає в тому, щоб спочатку впорядкувати всі елементи за молодшим розрядом, потім стабільно впорядкувати за другим розрядом, потім за третім і так далі аж до найстаршого. Оскільки, припускається, що кожен розряд приймає значення з невеликого діапазону, то кожен цикл впорядкування можна виконувати швидко і з малими затратами пам'яті.

Клас складності обох алгоритмів O(n) бо там є головний цикл по n

Проги:

lab\_5\_1\_Bucket\_Sort\lab\_5\_1\_Bucket\_Sort.sln і

lab\_5\_2\_RadixSort\lab\_5\_2\_RadixSort.sln написані на C#

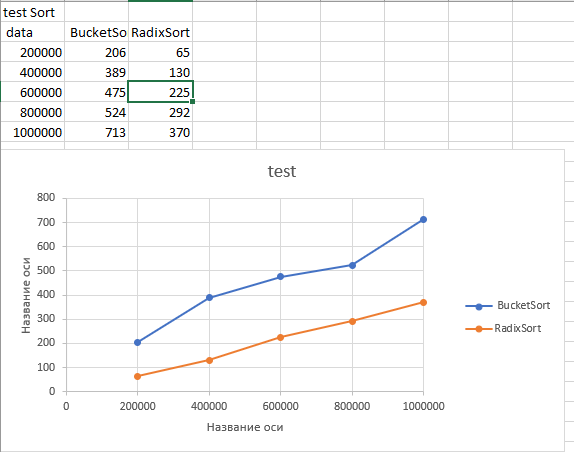
прога запускає цикли для різних даних і зберігає час виконання в файлі

lab\_5\_1\_Bucket\_Sort\bin\Debug\data1.csv

lab\_5\_2\_RadixSort\bin\Debug\data1.csv

графіки є в файлі data5.xlsx

Файл для тестування алгоритмів з невеликою кількістю даних: \bin\Debug\data2.csv



Висновок: в С# блочне і порозрядне сортування працюють з лінійною складністю для рівномірно розподілених даних, складність алгоритму в середньому O(n). Але порозрядне сортування в найгiршому випадку має O(n), а блочне сортування O(n^2).

