

Лабораторная работа № 1 по курсу дискретного анализа: сортировка за линейное время

Выполнил студент группы 08-208 МАИ *Левитанов Денис*.

Условие

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.
2. Карманная сортировка. Тип ключей: вещественные числа в промежутке $[-100, 100]$, тип значений: строки переменной длины (до 2048 символов).

Метод решения

Считываются пары «ключ значение» в структуру, эта структура помещается в вектор(собственная реализация). После считывания всех данных, вектор сортируется: создается вектор, состоящий из других векторов, размером равным количеству введенных элементов. Каждый элемент добавляется в вектор под индексом $[\text{ключ} + 100(\text{длина свига с } -100;100 \text{ на } 0;200))/201(\text{количество элементов на отрезке } -100;100) * \text{количество введенных элементов}]$, тем самым смещая данный отрезок $-100;100$ на отрезок $0;1$, который требуется для карманной сортировки. Затем каждый "карман" вектора сортируется вставкой, после всё соединяется в один вектор и выводится на экран.

Описание программы

1. *Заголовочные файлы:*
EVector.h(описание вектора)
Element.h(структура с полями double Key, char* Value)
BucketSort.h(Объявление карманной сортировки).
2. *Cpp файлы:*
main.cpp(Считывание и вывод данных, управление программой)
BucketSort.cpp (Реализация карманной сортировки, сортировка вставкой)
TVector.cpp(Реализация вектора).

Дневник отладки

Программа не проходила тест по памяти из-за считывания строки переменной длины в переменную с выделенной памятью в 2048 символов. Решение было найдено такое, что строка считывается во временную переменную с выделенной памятью в 2048 символов, затем считается длина этой строки, в структуре выделяется для неё память и строка копируется в структуру.

Тест производительности

1. 100 000 элементов:
Time of working bucket: 0.062256
Time of working std: 0.077026
2. 1 000 000 элементов:
Time of working bucket: 0.75178
Time of working std: 0.966112
3. 5 000 000 элементов:
Time of working bucket: 5.59361
Time of working std: 7.78683

Выводы

Карманная сортировка применяется только в тех случаях когда известен сортируемый отрезок и элементы равномерно распределены на этом отрезке.