**Практическое задание по теме «Алгоритмы сортировок»**

Общее задание

1) Реализовать на языке С++ сортировки для массива объектов в соответствии с вариантом.

2) Перегрузить операторы сравнения (>, <, >=, <=) для сравнения объектов.

3) Входные данные для сортировки массива обязательно считывать из внешних источников: текстовый файл, файл MS Excel, MS Access, данные из СУБД (любое на выбор). Выходные данные (отсортированный массив) записывать в файл.

4) Выбрать 7-10 наборов данных для сортировки размерности от 100 и более (но не менее 100000). Засечь (программно) время сортировки каждым алгоритмом. По полученным точкам построить графики зависимости времени сортировки от размерности массива для каждого из алгоритмов сортировки на одной оси координат. Сделать вывод о том, в каком случае, какой из методов лучше применять. Графики можно строить программно или в любой из прикладных программ (MS Excel, Matlab, MathCad и т.д.).

5) Сделать отчет, состоящий из:

* документацию к коду работы, сгенерированную с помощью case средства (doxygen, sphinx, etc);
* ссылку на исходный код программы в репозитории;
* графики времени сортировок.

*Варианты сортировок:*

а) Сортировка выбором

б) Сортировка пузырьком

в) Сортировка простыми вставками

г) Шейкер-сортировка

д) Пирамидальная сортировка

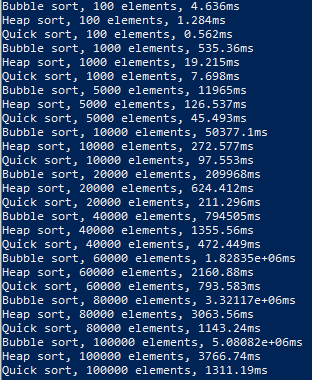
е) Быстрая сортировка

ж) Сортировка слиянием

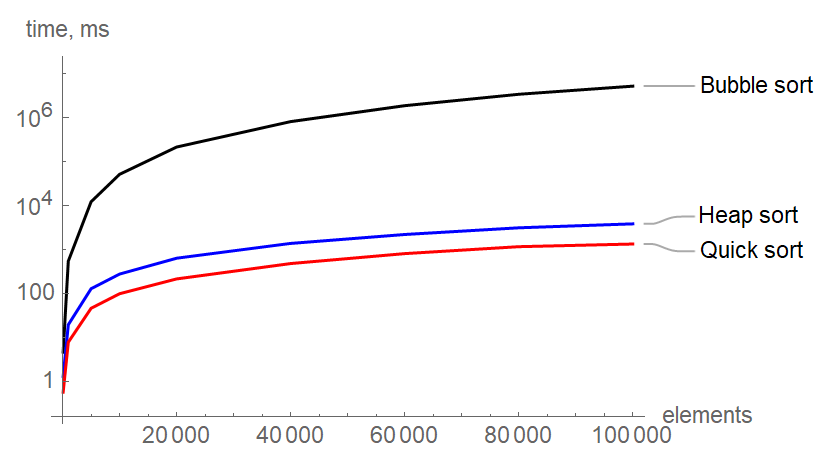
|  |  |
| --- | --- |
| 2 | Массив данных библиотечных книг: ФИО автора, название, год издания, количество страниц *(сравнение по полям - ФИО автора, название, год издания)* |

Ссылка на репозиторий:

Время работы сортировок:



Графики времени работы сортировок:



Вывод: т.к. свойства пирамидальной и быстрой сортировок схожи, то из-за времени предпочтительнее использовать быструю сортировку в случае, если нам не важна устойчивость сортировки. Если устойчивость сортировки важна, то следует применять сортировку пузырьком.

Ссылка на репозиторий: https://github.com/nbs13372/lab1.git