
Лабораторная работа №2

по курсу «Информатика (организация и поиск данных)» (3 семестр)

Варианты заданий

Постановка задачи

Исходной для данной задачи является реализация класса `Sequence<T>`, полученная в рамках курса за 2-й семестр, а также реализация умных указателей, полученные в ЛР-1 в 3-м семестре. Требуется написать программу на C++ для сравнения различных алгоритмов сортировки. Написать краткое техническое задание (ТЗ). Выполнить реализацию. Написать для нее тесты.

Минимальные требования к программе. В программе должно быть реализовано не менее 2-х различных алгоритмов (см. табл. «Выбор вариантов» ниже). Основные алгоритмы необходимо покрыть тестами. Это касается и реализованного ранее типа `Sequence`¹. Программа должна позволять выбрать любой из реализованных алгоритмов сортировки и запустить его на (достаточно произвольных) исходных данных. Алгоритмы сортировки должны быть параметризованы способом сравнения элементов. При этом должна быть возможность как автоматической, так и ручной проверки корректности работы алгоритмов (в т.ч. должна быть возможность просмотра как исходных данных, так и результата – с помощью вывода на экран или/и вывода в файл). Программа должна обладать пользовательским интерфейсом (консольным или графическим). Программа должна позволять проводить проверку работы алгоритмов на длинных последовательностях (10 000 элементов и более); должна быть возможность автоматической генерации последовательностей заданной длины (например, с помощью генератора случайных чисел). Пользовательский интерфейс, в особенности, графический, тестировать не требуется. Программа должна предоставлять функцию измерения времени выполнения алгоритма. Должна быть функция сравнения алгоритмов – по времени выполнения на одних и тех же входных данных².

Методические указания. Реализовать абстрактный тип данных – последовательность. Выполнить реализацию в 2-х вариантах – на основе массивов и на основе связанных списков. Полученные результаты пригодятся в последующих заданиях. Алгоритмы сортировки могут быть реализованы как в виде отдельных функций, так и инкапсулированы в класс, унаследованный от чисто абстрактного класса (интерфейса) `ISorter`. Сигнатура функции сортировки:

```
Template<typename T>  
Sequence<T>* Sort (Sequence<T>* seq, int (*cmp) (T,T));
```

¹ По меньшей мере, должны быть тестами те методы, которые использованы в рамках данной лабораторной работы.

² Следует рассматривать три основных случая: массив уже отсортирован в нужном направлении; массив отсортирован в обратном направлении; массив не отсортирован.

Выбор варианта задания

Каждый студент должен выбрать для реализации 2-3 алгоритма сортировки, исходя из того, чтобы сумма баллов за все выбранные алгоритмы была не менее 14. В таблице ниже приведен список алгоритмов и соответствующие им баллы; количество баллов пропорционально сложности алгоритма.

№	Название алгоритма	Кол-во баллов
1.	Метод пузырька	3
2.	Модификация метода пузырька – шейкерная сортировка	5
3.	Метод простых вставок	5
4.	Сортировка с помощью простого выбора	5
5.	Сортировка подсчетом	7
6.	Метод двоичных вставок	7
7.	Квадратичная сортировка (усов. сортировка выбором)	9
8.	Сортировка с помощью выбора из дерева	7
9.	Сортировка слиянием	7
10.	Пирамидальная сортировка	10
11.	Быстрая сортировка	10
12.	Сортировка Шелла	10
13.	Сортировка Шелла (с выбором смещения)	11
14.	Схема Бэтчера	11

Пример. При выборе метода пузырька, простых вставок и быстрой сортировки сумма баллов: $3+7+10=20$.

Критерии оценки

1.	Качество программного кода:	<ul style="list-style-type: none">– стиль (в т.ч.: имена, отступы и проч.) (0-2)– структурированность (напр. декомпозиция сложных функций на более простые) (0-2)– качество основных и второстепенных алгоритмов (напр. обработка граничных случаев и некорректных исходных данных и т.п.) (0-3)	0-7 баллов
2.	Качество пользовательского интерфейса:	<ul style="list-style-type: none">– предоставляемые им возможности (0-2)– наличие ручного/автоматического ввода исходных данных (0-2)– настройка параметров для автоматического режима отображение исходных данных и промежуточных и конечных результатов и др. (0-2)	0-6 баллов
3.	Качество тестов	<ul style="list-style-type: none">– степень покрытия– читаемость	0-4 баллов

		– качество проверки (граничные и некорректные значения, и др.)	
4.	Полнота выполнения задания и качество ТЗ	Оценивается качество подготовки ТЗ, полнота выполнений минимальных требований	0-5 баллов
5.	Владение теорией	знание алгоритмов, области их применимости, умение сравнивать с аналогами, оценить сложность, корректность реализации	0-3 баллов
6.	Оригинальность реализации	оцениваются отличительные особенности конкретной реализации – например, общность структур данных, наличие продвинутых графических средств, средств ввода-вывода, интеграции с внешними системами и др.	0-9 баллов
Итого			0-34 баллов

Для получения зачета за выполнения лабораторной работы необходимо соблюдение всех перечисленных условий:

- оценка за п. 1 должна быть не менее 3 баллов
- оценка за п. 4 должна быть не менее 3 баллов
- оценка за п. 5 должна быть больше 0
- суммарная оценка за работу без учета п. 6 должна быть не менее 15 баллов