

Лабораторная работа № 1 “Исследование простых алгоритмов сортировки”

Задание:

1. Разработать проект для исследования алгоритмов сортировки в соответствии с вариантом (диаграмма вариантов использования проекта представлена на рис. 1)

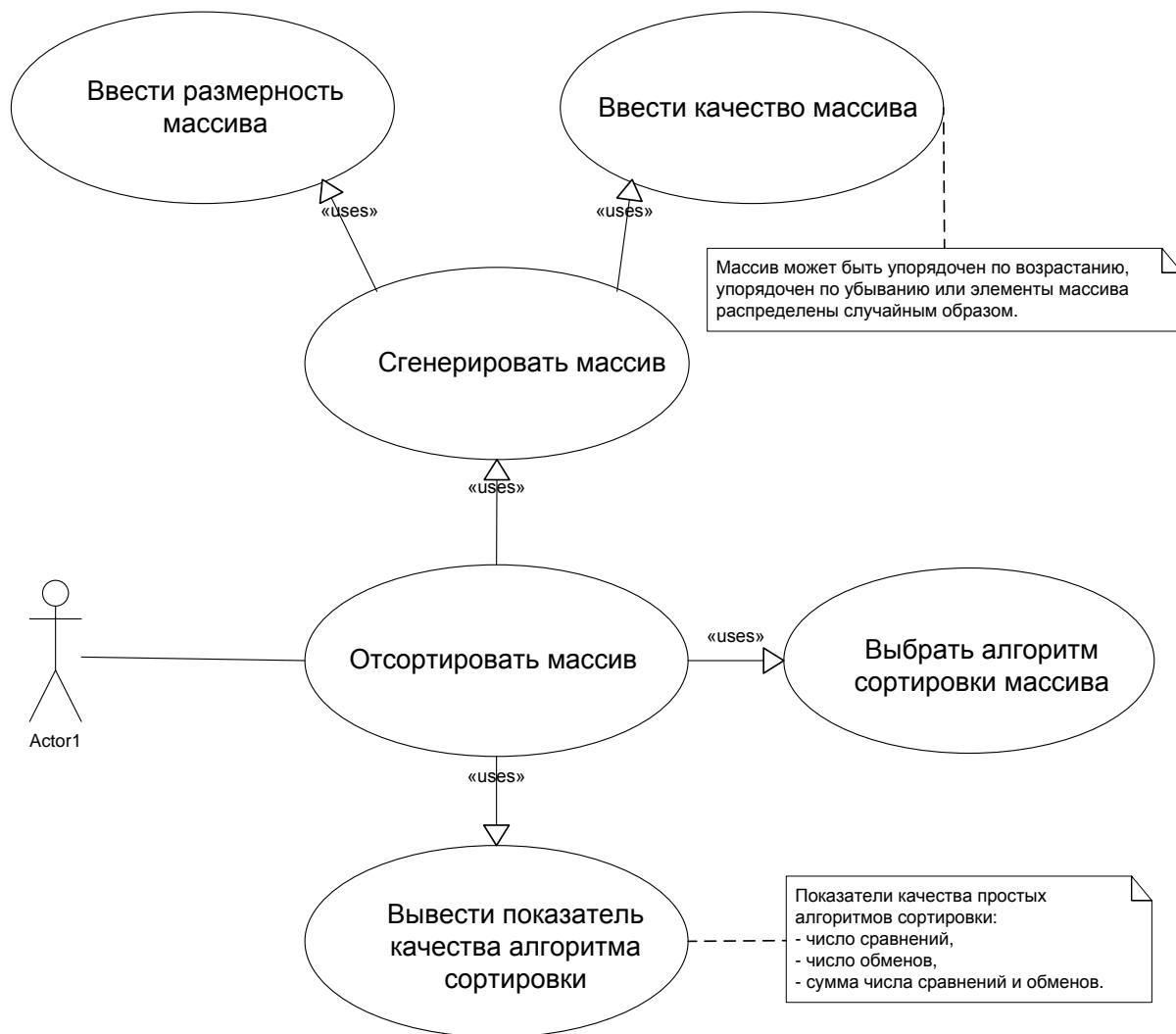


Рис. 1. Диаграмма вариантов использования проекта лабораторной работы № 1.

2. Разработать интерфейс проекта, *позволяющий*:

- задавать размерность и качество массива;
- осуществлять выбор алгоритма сортировки для исследования;
- осуществлять вывод информации о результатах исследования алгоритма сортировки (исходный и отсортированный массив, пошаговую

работу алгоритма сортировки (при небольшой размерности массива), показатели качества работы алгоритма сортировки).

3. Создать подпрограмму для генерации целочисленного массива различного качества. В подпрограмме *предусмотреть*:

- задание размерности массива;
- задание диапазона чисел массива;
- создание массива с равномерно распределенными случайными числами;
- создание упорядоченного по возрастанию массива;
- создание упорядоченного по убыванию массива.

4. Создать подпрограмму, реализующую алгоритм сортировки в соответствии с вариантом. В подпрограмме *предусмотреть*:

- определение числа сравнений;
- определение числа обменов;
- определение суммы обменов и сравнений;
- сортировку массива по возрастанию.

Письменный отчет по лабораторной работе должен содержать:

1. Титульный лист. (Название лабораторной работы. Фамилия, имя, отчество, номер группы исполнителя, дата сдачи.)
2. Математическую постановку задачи.
3. Распечатку текстов подпрограмм для генерации массива и для исследования алгоритмов сортировки (обязательны комментарии к программе).
4. Исследование программной реализации простых алгоритмов сортировки, содержащее следующие материалы, таблицы и графики:
 - *примеры пошаговой работы* исследуемых алгоритмов сортировки для небольшой размерности задачи ($n = 7$, диапазон чисел 0..10);
 - сведенную в *таблицу* зависимость количества сравнений, обменов (присваиваний) и их суммы (или времени выполнения) исследуемых алгоритмов сортировки от размерности задачи n для прямого, обратного и случайного расположения элементов в массиве (всего в отчете должно быть 6 таблиц);

Пример таблицы:

Таблица 1 Сортировка простой вставкой, массив упорядочен по возрастанию

n	5	10	15	20	25	30	35	40	45	50
Sr										
Ob										
Sum										

- *графики* зависимости показателя качества (количества операций (суммы числа сравнений и обменов) или времени выполнения)

исследуемых алгоритмов сортировки от размерности задачи для прямого, обратного и случайного расположения элементов в массиве (всего в отчете должно быть 5 графиков);

5. *Выводы* по лабораторной работе (в выводах провести сравнительную характеристику исследованных алгоритмов сортировки).

Варианты задач по лабораторной работе:

1. Сортировка простыми включениями (простой вставкой).
2. Сортировка методом пузырька (простым обменом).
3. Шейкер-сортировка.
4. Сортировка простым выбором.

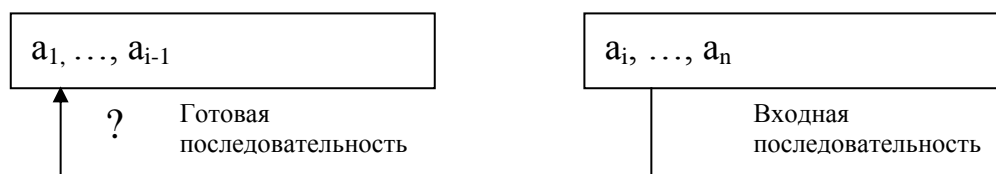
Варианты исследуемых алгоритмов сортировки

Вариант	Интервал $[A, B]$	Варианты задач
1.	[0,100]	1, 2
2.	[0,200]	1, 3
3.	[0,200]	1, 4
4.	[100,300]	2, 4
5.	[0,500]	1, 2
6.	[100,200]	1, 3
7.	[0,100]	1, 4
8.	[0,200]	2, 4
9.	[0,300]	1, 2
10.	[0,400]	1, 3
11.	[200,500]	1, 4
12.	[0,600]	2, 4
13.	[0,700]	1, 2
14.	[0,800]	1, 3
15.	[100,900]	1, 4
16.	[0,999]	2, 4
17.	[1,900]	1, 2
18.	[0,1000]	1, 3
19.	[0,100]	1, 4
20.	[10,100]	2, 4

Краткие теоретические сведения

1. Сортировка простыми включениями (вставками).

Элементы условно разделяются на готовую (упорядоченную) последовательность a_1, \dots, a_{i-1} и входную последовательность a_i, \dots, a_n .



На каждом шаге сортировки, начиная с $i=2$ и увеличивая i на 1, берут i -й элемент входной последовательности и пересылают его в готовую последовательность, вставляя (включая) его на подходящее место.

Процесс сортировки простыми включениями для последовательности (7, 2, 1, 8, 3) показан в табл. 1.

Таблица 1. Пример сортировки простыми включениями

Индекс i	Состояние последовательности					Сравнений	Обменов	Сумма
$i=2$	2	7	1	8	3	1	1	2
$i=3$	1	2	7	8	3	2	2	4
$i=4$	1	2	7	8	3	1	0	1
$i=5$	1	2	3	7	8	3	2	5

Таким образом, для сортировки простыми включениями последовательности (7, 2, 1, 8, 3) потребовалось 12 операций (сравнений и обменов).

При пересылке (вставке) очередного элемента x в уже отсортированную часть последовательности выполняется чередование сравнений и присваиваний (пересылок), то есть готовую последовательность “просеивают” по x . Поиск места вставки может закончиться в двух случаях, если:

- 1) найден элемент a_j , меньший, чем x ;
- 2) достигнут левый конец готовой последовательности.

Для контроля достижения левого конца готовой последовательности может использоваться фиктивный элемент (“барьер”), добавляемый слева к готовой последовательности, его необходимо учесть при описании нижней границы индекса массива, используемого для размещения сортируемой последовательности.

Выводы по алгоритму сортировки простыми включениями:

- 1) временная сложность алгоритма равна $O(n^2)$;
- 2) у алгоритма естественное поведение, так как максимальное количество сравнений и присваиваний выполняется в наихудшем случае, когда исходная последовательность упорядочена в обратном порядке;
- 3) алгоритм описывает устойчивую сортировку, так как оставляет неизменным порядок элементов с одинаковыми значениями.

2. Сортировка методом пузырька (простым обменом).

Сортировка простым обменом основана на сравнении и при необходимости обмене пар соседних элементов до тех пор, пока не будут упорядочены все элементы сортируемой последовательности – элементы, расположение которых не соответствует порядку, меняются местами, обмен выполняется до тех пор, пока остаются такие пары. Начинить сравнение и при необходимости обмен элементов можно с любого конца массива.

Если мы будем рассматривать массив элементов, расположенных вертикально и представим себе элементы пузырьками в воде, обладающими “весами”, соответствующими их ключам, то каждый проход по массиву приводит к “всплыванию” пузырька на соответствующий его весу уровень. Этот метод широко известен как сортировка методом пузырька.

Процесс сортировки простым обменом для последовательности (7, 2, 1, 8, 3) показан в табл. 2.

Таблица 2.

Пример сортировки простым обменом

	$i=1$	$i=2$	$i=3$	$i=4$
	7	1	1	1
	2	7	2	2
	1	2	7	3
	8	3	3	7
	3	8	8	8
<i>Сравнений</i>	4	3	2	1
<i>Обменов</i>	3	1	1	0
<i>Сумма</i>	7	4	3	1

Таким образом, для сортировки простым обменом последовательности (7, 2, 1, 8, 3) потребовалось 15 операций (сравнений и обменов).

Выводы по алгоритму сортировки простым обменом:

- 1) временная сложность $O(n^2)$.
- 2) имеет место естественное поведение алгоритма, так как количество присваиваний максимально в наихудшем случае;
- 3) основной недостаток алгоритма – большое количество присваиваний.

3. Шейкер-сортировка.

Алгоритм сортировки простым обменом можно улучшить, используя следующие соображения. Если в процессе сортировки при каком-либо проходе нет обменов, то это значит, что последовательность отсортирована и сортировку можно остановить.

Шейкер-сортировка отличается от сортировки методом пузырька тем, что в ней чередуются проходы снизу-вверх и сверху- вниз.

В основу алгоритма шейкер-сортировки положено:

- запоминание места последнего обмена;
- чередование направлений проходов вперед – назад.

Процесс шейкер-сортировки для последовательности (7, 2, 1, 8, 3) показан в табл. 3.

Таблица 3.

Пример шейкер-сортировки

	$i=1$	$i=2$	$i=3$	$i=4$
	7	1	1	1
	2	↓ 7	2	2
	1	2	3	↓ 3
	8	3	↑ 7	7
	↑ 3	8	8	8
<i>Сравнений</i>	4	3	2	1
<i>Обменов</i>	3	2	0	0

<i>Сумма</i>	<i>7</i>	<i>5</i>	<i>2</i>	<i>1</i>
--------------	----------	----------	----------	----------

Таким образом, для шейкер-сортировки последовательности (7, 2, 1, 8, 3) потребовалось 15 операций (сравнений и обменов).

Для алгоритма шейкер-сортировки делаем следующие выводы:

- 1) временная сложность алгоритма $O(n^2)$;
- 2) имеет место естественное поведение алгоритма;
- 3) его недостаток (как и у метода простого обмена) – на каждом элементарном шаге сортировки элементы перемещаются только на одну позицию, что в общем случае приводит к большому количеству присваиваний.

4. Сортировка простым выбором.

Идея алгоритма сортировки простым выбором состоит в следующем:

- 1) найти наименьший элемент, исследуя всю последовательность;
- 2) поменять местами найденный наименьший элемент с первым;
- 3) повторить пункты 1, 2 на остальных $(n-1)$, $(n-2)$, ..., 2 элементах последовательности.

Процесс сортировки простым выбором для последовательности (7, 2, 1, 8, 3) показан в табл. 4.

Таблица 4. Пример сортировки простым выбором

<i>Индекс i</i>	<i>Состояние последовательности</i>					<i>Сравнений</i>	<i>Обменов</i>	<i>Сумма</i>
$i=1$	1	2	7	8	3	4	1	5
$i=2$	1	2	7	8	3	3	1	4
$i=3$	1	2	3	8	7	2	1	3
$i=4$	1	2	3	7	8	1	1	2

Таким образом, для сортировки простым выбором последовательности (7, 2, 1, 8, 3) потребовалось 14 операций (сравнений и обменов).

При первом выполнении цикла сортировки выполняется $(n-1)$ сравнений и задача размера n сводится к задаче размером $(n-1)$. Далее задача размером $(n-1)$ сводится к задаче размером $(n-2)$ и так далее.

Алгоритм сортировки простым выбором имеет временную сложность $T(n)=O(n^2)$.

Для алгоритма сортировки простым выбором справедливо следующее:

- 1) временная сложность алгоритма $O(n^2)$;
- 2) имеет место естественное поведение алгоритма;
- 3) отсутствие сбалансированности - используется разбиение задачи на неравные части.

Метод сортировки простым выбором в некотором смысле противоположен сортировке простыми включениями; при сортировке простыми включениями на каждом шаге рассматривается только один очередной элемент входной последовательности и все элементы готового массива для нахождения места включения; при сортировке простым выбором рассматриваются все элементы с наименьшим значением, и этот один очередной элемент отправляется в готовую последовательность.

Таким образом, обычно сортировка простым выбором предпочтительнее сортировки простыми включениями, хотя в случае, когда ключи заранее отсортированы или почти отсортированы, сортировка простыми включениями все же работает несколько быстрее.

Список литературы

1. Вирт Н. Алгоритмы и структуры данных.- М.: Мир, 1989.
2. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы.- М.: Вильямс, 2000.
3. Лэнгсам И., Огенстайн М., Тененбаум А. Структуры данных для персональных ЭВМ. -М.: Мир, 1989.
4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ.-М.: МЦНМО, 2000.
5. Кнут Д. Искусство программирования для ЭВМ. Т.3.Сортировка и поиск. -М.: Мир, 1979.
6. Зубов В.С. Справочник программиста. Базовые методы решения графовых задач и сортировки.- М.:Филинь, 1999.
7. Макконелл Дж. Анализ алгоритмов. Вводный курс.-М.: Техносфера, 2002.

Контрольные вопросы и задания

1. Дайте определение понятию “алгоритм”.
2. Какими особенностями должен обладать алгоритм?
3. Дайте определение временной сложности алгоритма.
4. Каковы основные правила работы с *O*-символикой?
5. Каковы основные этапы полного построения алгоритма?
6. В чем состоят задача и цель сортировки?
7. Дайте классификацию алгоритмов внутренней сортировки.
8. Каковы основные показатели качества алгоритма сортировки?
9. Отсортируйте пошагово по возрастанию массив целых чисел размерности $n=6$ с помощью сортировки простой вставкой, сортировки простым обменом, шейкер-сортировки и сортировки простым выбором.
10. Каковы сложности алгоритмов сортировки простой вставкой, сортировки простым обменом, шейкер-сортировки и сортировки простым выбором?
11. Как задается качество массива для исследования алгоритмов сортировки?