

Задание по теме «Сортировки в Си»

1. Необходимо отсортировать первые две трети массива в порядке возрастания если среднее арифметическое всех элементов больше нуля; иначе - лишь первую треть. остальную часть массива не сортировать, а расположить в обратном порядке.

2. Написать программу, реализующую сортировку массива методом усовершенствованной сортировки пузырьковым методом. Усовершенствование состоит в том, чтобы анализировать количество перестановок на каждом шагу, если это количество равно нулю, то продолжать сортировку нет смысла – массив отсортирован.

3. Есть стопка оладий различного радиуса. Единственная операция, проводимая с ними - между любыми двумя сует лопатку и меняем порядок оладий над лопаткой на обратный. Необходимо за минимальное количество операций таких отсортировать снизу-вверх по убыванию радиуса.

4. Написать программу, сортирующую методом вставок двумерный массив целых чисел. При этом самое маленькое число должно оказаться в первом элементе первой строки, а самое большое в последнем элементе последней строки.

5. Написать программу, сортирующую по алфавиту строку латинских букв

6. Написать программу, сортирующую список студентов группы по алфавиту

7. Сортировка «с конфеткой» выполняется по тем же правилам, но дополнительно задан список пар чисел, которые не меняются друг с другом ни при каких условиях (в таком случае сортирующий получает конфетку за то, что пропускает соответствующий обмен). Например, наличие в списке пары (4,1) обозначает, что если в какой-то момент рядом окажутся числа 4 и 1, и по алгоритму сортировки их нужно будет поменять местами, то обмена не произойдет, а сортирующий получит конфетку.

Требуется провести сортировку «с конфеткой» данного массива и выдать результат сортировки.

8. На планете «Аурон» атмосфера практически отсутствует, поэтому она известна своими перепадами температур в различных точках. Известно, что эти перепады колеблются от -100 до 100 градусов. Нашим специалистам удалось выяснить значения температур в N точках этой планеты. К сожалению, эти значения вычислены с большими погрешностями, поэтому их решили округлить до целых чисел. Хотелось бы наглядно видеть участки с повышенной и пониженной температурой. Вам требуется помочь. Вы должны упорядочить температуры участков по возрастанию.

9. В файле данных хранится информация о каждом из воспитанников детского сада: фамилия, имя, название группы (младшая средняя старшая), заключение о состоянии здоровья ("здоров" или "нуждается в лечении") каждого из четырёх специалистов - невропатолога, отоларинголога, ортопеда и стоматолога. Разработать и реализовать программу "Учёт результатов диспансеризации", которая считывает исходную информацию и позволяет организовать обработку следующих запросов:

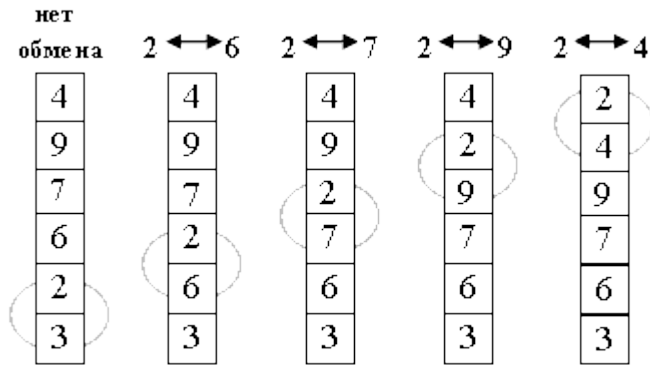
- Определить процентное соотношение здоровых и нуждающихся в лечении детей по заключению указанного в запросе специалиста;
- Вывести на экран, в удобной для пользователя форме полную информацию по указанной в запросе группе. При этом сведения должны быть упорядочены в алфавитном порядке.
- Число воспитанников детского сада не превышает 100.

Методы сортировки

Сортировка пузырьковым методом

Расположим массив сверху вниз, от нулевого элемента - к последнему.

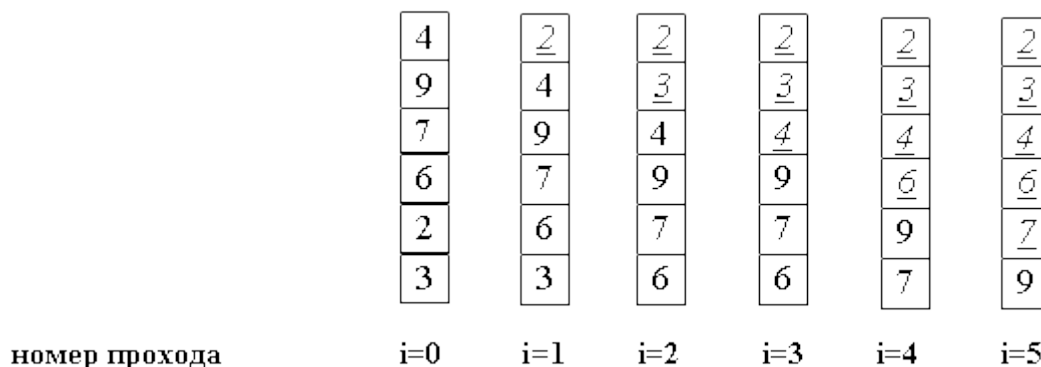
Идея метода: шаг сортировки состоит в проходе снизу вверх по массиву. По пути просматриваются пары соседних элементов. Если элементы некоторой пары находятся в неправильном порядке, то меняем их местами.



Нулевой проход, сравниваемые пары выделены

После нулевого прохода по массиву "вверху" оказывается самый "легкий" элемент - отсюда аналогия с пузырьком. Следующий проход делается до второго сверху элемента, таким образом второй по величине элемент поднимается на правильную позицию...

Делаем проходы по все уменьшающейся нижней части массива до тех пор, пока в ней не останется только один элемент. На этом сортировка заканчивается, так как последовательность упорядочена по возрастанию.



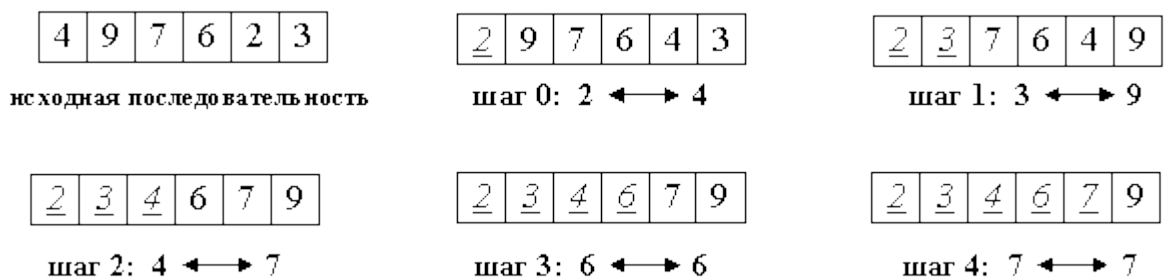
Среднее число сравнений и обменов имеют квадратичный порядок роста: (n^2) , отсюда можно заключить, что алгоритм пузырька очень медленен и малоэффективен.

Сортировка выбором

Идея метода состоит в том, чтобы создавать отсортированную последовательность путем присоединения к ней одного элемента за другим в правильном порядке.

Будем строить готовую последовательность, начиная с левого конца массива. Алгоритм состоит из n последовательных шагов, начиная от нулевого элемента и заканчивая $(n-1)$ -м.

На i -м шаге выбираем наименьший из элементов $a[i] \dots a[n]$ и меняем его местами с $a[i]$. Последовательность шагов при $n=5$ изображена на рисунке ниже.



Вне зависимости от номера текущего шага i , последовательность $a[0] \dots a[i]$ является упорядоченной. Таким образом, на $(n-1)$ -м шаге вся последовательность, кроме $a[n]$ оказывается отсортированной, а $a[n]$ стоит на последнем месте по праву: все меньшие элементы уже ушли влево.

Для нахождения наименьшего элемента из $n+1$ рассматриваемых алгоритм совершает n сравнений. С учетом того, что количество рассматриваемых на очередном шаге элементов уменьшается на единицу, общее количество операций:

$$n + (n-1) + (n-2) + (n-3) + \dots + 1 = 1/2 * (n^2 + n) = (n^2)/2.$$

Таким образом, так как число обменов всегда будет меньше числа сравнений, время сортировки растет квадратично относительно количества элементов.

Сортировка вставками

Сортировка простыми вставками в чем-то похожа на сортировку выбором и сортировку пузырьковым методом.

Аналогичным образом делаются проходы по части массива, и аналогичным же образом в его начале "вырастает" отсортированная последовательность...

Однако в сортировке пузырьком или выбором можно было четко заявить, что на i -м шаге элементы $a[0]...a[i]$ стоят на правильных местах и никуда более не переместятся. Здесь же подобное утверждение будет более слабым: последовательность $a[0]...a[i]$ упорядочена. При этом по ходу алгоритма в нее будут вставляться (см. название метода) все новые элементы.

Будем разбирать алгоритм, рассматривая его действия на i -м шаге. Как говорилось выше, последовательность к этому моменту разделена на две части: готовую $a[0]...a[i]$ и неупорядоченную $a[i+1]...a[n]$.

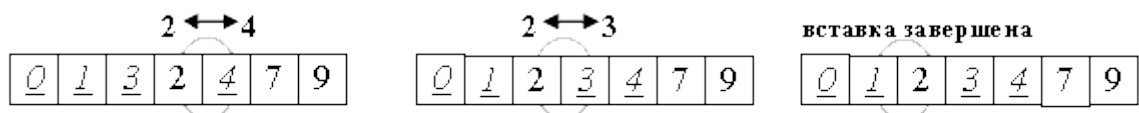
На следующем, $(i+1)$ -м каждом шаге алгоритма берем $a[i+1]$ и вставляем на нужное место в готовую часть массива.

Поиск подходящего места для очередного элемента входной последовательности осуществляется путем последовательных сравнений с элементом, стоящим перед ним.

В зависимости от результата сравнения элемент либо остается на текущем месте (вставка завершена), либо они меняются местами и процесс повторяется.

<u>0</u>	<u>1</u>	<u>3</u>	<u>4</u>	2	7	9
----------	----------	----------	----------	---	---	---

Последовательность на текущий момент. Часть $a[0]...a[2]$ уже упорядочена.



Вставка числа 2 в отсортированную подпоследовательность. Сравниваемые пары выделены.

Таким образом, в процессе вставки мы "просеиваем" элемент x к началу массива, останавливаясь в случае, когда

1. Найден элемент, меньший x или
2. Достигнуто начало последовательности.

Аналогично сортировке выбором, среднее, а также худшее число сравнений и пересылок оцениваются как (n^2) , дополнительная память при этом не используется.

Сортировка Шелла

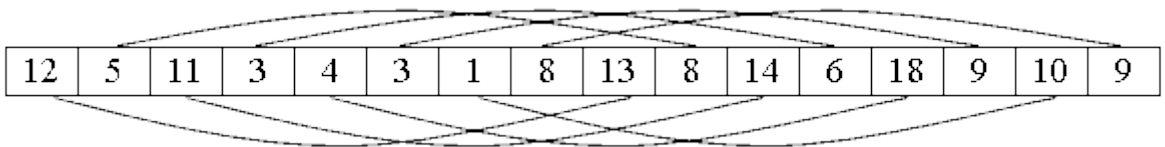
Сортировка Шелла является довольно интересной модификацией алгоритма сортировки простыми вставками.

Рассмотрим следующий алгоритм сортировки массива $a[0]..a[15]$.

12	8	14	6	4	9	1	8	13	5	11	3	18	3	10	9
----	---	----	---	---	---	---	---	----	---	----	---	----	---	----	---

1. Вначале сортируем простыми вставками каждые 8 групп из 2-х элементов ($a[0]$, $a[8]$), ($a[1]$, $a[9]$), ... , ($a[7]$, $a[15]$).

12	5	11	3	4	3	1	8	13	8	14	6	18	9	10	9
----	---	----	---	---	---	---	---	----	---	----	---	----	---	----	---



2. Потом сортируем каждую из четырех групп по 4 элемента ($a[0]$, $a[4]$, $a[8]$, $a[12]$), ..., ($a[3]$, $a[7]$, $a[11]$, $a[15]$).

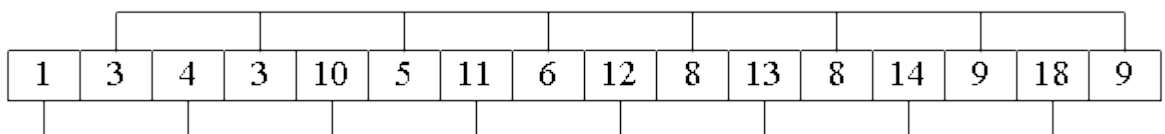
номер группы:

0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
4	3	1	3	12	5	10	6	13	8	11	8	18	9	14	9

В нулевой группе будут элементы 4, 12, 13, 18, в первой - 3, 5, 8, 9 и т.п.

3. Далее сортируем 2 группы по 8 элементов, начиная с ($a[0]$, $a[2]$, $a[4]$, $a[6]$, $a[8]$, $a[10]$, $a[12]$, $a[14]$).

1	3	4	3	10	5	11	6	12	8	13	8	14	9	18	9
---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---



4. В конце сортируем вставками все 16 элементов.

1	3	3	4	5	6	8	8	9	9	10	11	12	13	14	18
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

Очевидно, лишь последняя сортировка необходима, чтобы расположить все элементы по своим местам. Так зачем нужны остальные?

На самом деле они продвигают элементы максимально близко к соответствующим позициям, так что в последней стадии число перемещений

будет весьма невелико. Последовательность и так почти отсортирована. Ускорение подтверждено многочисленными исследованиями и на практике оказывается довольно существенным.

Единственной характеристикой сортировки Шелла является *приращение* - расстояние между сортируемыми элементами, в зависимости от прохода. В конце приращение всегда равно единице - метод завершается обычной сортировкой вставками, но именно последовательность приращений определяет рост эффективности.

Среднее количество операций: $(n^{7/6})$, в худшем случае - порядка $(n^{4/3})$.