



Урок 20. C++: Сортировка пузырьком

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 20 из 34 · 45 минут · Модуль «Сортировка»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Соседние обмены постепенно перемещают большой элемент вправо. В C++ важно не обратиться к $a[j+1]$ за границей и не вычесть единицу из нулевой беззнаковой длины.

Результат урока: Реализовать пузырьковую сортировку `vector`, безопасно обработать пустоту и проверить раннюю остановку счётчиком сравнений.

01. Проблемная ситуация

0–5 минут. Ряд $[5, 1, 4, 2]$. Сделайте один проход слева направо, меняя соседей, если левый больше. Запишите ряд после каждого обмена. Готов ли весь ряд после попадания 5 на правый край?

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Граница прохода и ранняя остановка

Переменная `end` — число элементов ещё проверяемой части. Цикл $j+1 < end$ гарантирует допустимость обеих позиций. После полного прохода максимум этой части оказывается на `end-1`; следующая проверяемая часть короче на один. Условие $end > 1$ не даёт войти в цикл для размеров 0 и 1.

Если за проход не было обменов, все соседние пары текущей части уже не убывают. Вместе с готовым суффиксом это означает готовность всего ряда. В коде с `changed` только сравнения значений $a[j] > a[j+1]$, `s` — фактические обмены. Сравнения условий циклов сюда не входят.

В худшем случае $n(n-1)/2$ сравнений, время $O(n^2)$, вспомогательная память $O(1)$. Уже упорядоченный ряд из $n \geq 2$ элементов даёт $n-1$ сравнений и остановку. Равные значения не меняются местами, поскольку знак строгий. Учебный контракт: `vector` из $0 \dots 100$ целых, значения $-1000 \dots 1000$.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для $[5, 1, 4, 2]$ первый проход: $[1, 5, 4, 2] \rightarrow [1, 4, 5, 2] \rightarrow [1, 4, 2, 5]$. Второй даёт $[1, 2, 4, 5]$, третий проверяет пару 1,2 без обмена. Всего $s=3+2+1=6$, $c=4$.

```
#include <iostream>
#include <vector>
#include <utility>
std::pair<int,int> bubble(std::vector<int>& a) {
    int c=0, s=0;
    for(std::size_t end=a.size(); end>1; --end) {
        bool changed=false;
        for(std::size_t j=0; j+1<end; ++j) {
            ++c;
            if(a[j]>a[j+1]) {
                std::swap(a[j],a[j+1]);
                ++s; changed=true;
            }
        }
        if(!changed) break;
    }
    return {c,s};
}
int main() {
    std::vector<int> a{5,1,4,2};
    auto [c,s]=bubble(a);
    for(int x:a) std::cout<<x<<' ';
    std::cout<<"\ncomparisons="<<c<<" swaps="<<s<<"\n";
}
```



Вход	5 1 4 2
Проход end=4	1 4 2 5
Проход end=3	1 2 4 5
Проход end=2	1 2 4 5; обменов нет

Справа от черты — уже готовый суффикс после соответствующего прохода.

03. Практика 1. Проверяем материал

Считаем сравнения и обмены

14–25 минут. Меняйте только инициализацию `a` в `main`, компилируйте с предупреждениями.

Вход	Итог; <code>c</code> , <code>s</code>
{5,1,4,2}	1 2 4 5; 6,4
{ }; {7}	Пусто; 0,0 / 7; 0,0
{1,2,3,4}	1 2 3 4; 3,0
{3,2,1}; {2,2,1}	1 2 3; 3,3 / 1 2 2; 3,2

2 минуты — прогноз, 4 — шесть запусков, 3 — сверка счётчиков и границ `j`, 2 — журнал. Критерии: все результаты, отсутствие обращения к элементу пустого вектора, объяснение `end>1` и `j+1<end`.

04. Практика 2. Перенос способа

Меняем направление сортировки

25–33 минуты. Нужно невозрастание для {1,4,2,4}. За 2 минуты выберите единственное сравнение, за 4 измените копию функции и получите счётчики, за 2 проверьте {4,4,2,1} и {}. Критерии: три правильных итога и объяснение равенства.

05. Обсуждение и границы вывода

33–40 минут. Можно ли остановиться после первого обмена или после первой пары без обмена? За 2 минуты проверьте обе идеи на примере, за 3 сформулируйте достаточное условие, за 2 сравните с готовностью суффикса.

06. Индивидуальная проверка

40–45 минут. Для {2,1,3} укажите ряд после первого прохода, общее `c`, общее `s` и значение `end` при первом проходе. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для пустоты выводится пустая строка значений и нулевые счётчики. У $\{2,2,1\}$ равная пара 2,2 не обменивается, но две двойки проходят через 1: два обмена. Нельзя начинать без проверки с $\text{size}()-1$: при нулевом size беззнаковое вычитание даёт огромное значение.

Практика 2: разбор и критерии

Замените $a[j]>a[j+1]$ на $a[j]<a[j+1]$. Для $\{1,4,2,4\}$: 4 4 2 1, $c=6, s=4$; для $\{4,4,2,1\}$: $c=3, s=0$; для $\{\}$: $c=s=0$. Равные четвёрки не меняются напрямую, знак остаётся строгим.

Обсуждение: возможный вывод

Первая пара без обмена не говорит о следующей: $\{1,3,2\}$ ещё не готов. Первый обмен тоже не заканчивает сортировку: $\{3,2,1\}$ после него станет $\{2,3,1\}$. Достаточно полного прохода без обменов; готовый суффикс позволяет сократить границу, но не игнорировать оставшуюся часть.

Индивидуальная проверка: ответ и критерии

После первого прохода 1 2 3; $c=3$, поскольку второй проход проверит одну пару; $s=1$; сначала $\text{end}=3$. По одному баллу.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Компилятор G++ с поддержкой C++17 и редактор. Сборка: `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. Запуск `./lesson` в Linux/macOS или `.\lesson.exe` в Windows (при сборке задайте `-o lesson.exe`). Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 9 класс. Предпосылки: `vector`, циклы, индексы, ссылки и `std::swap`. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Алгоритмы сортировки	Считаем сравнения и обмены	Реализовать пузырьковую сортировку vector, безопасно обработать пустоту и проверить раннюю остановку счётчиком сравнений.
Структурирование решения и отладка	Меняем направление сортировки	Исходник, журнал обычных и граничных тестов, самостоятельное объяснение

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

NIST: пузырьковая сортировка

<https://xlinux.nist.gov/dads/HTML/bubblesort.html>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.