



Урок 22. C++: Сортировка слиянием

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 22 из 34 · 45 минут · Модуль «Сортировка»

Уровень: начинающий программист; C++17.

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

Один общий буфер помогает объединять части массива. Самая частая ошибка — перепутать включённую левую границу и не включённую правую.

Результат урока: Реализовать слияние на полуинтервале $[l, r)$, проверить границы и сохранить элементы вне сортируемого участка.

01. Проблемная ситуация

0–5 минут. Вектор $[9, 4, 1, 3, 8]$, участок $[1, 4)$. Выпишите индексы, которые входят в участок, и ожидаемый вектор после сортировки только этого участка. Должны ли двигаться 9 и 8?

02. Понятия и разобранный пример

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

Полуинтервал и общий буфер

Полуинтервал $[l, r)$ включает l и не включает r , его длина $r-l$. База $r-l < 2$ покрывает пустоту и один элемент. Середина $m = l + (r-l)/2$ делит участок на $[l, m)$ и $[m, r)$, которые не пересекаются и покрывают исходный участок.

Два индекса i, j читают уже отсортированные половины, k пишет в буфер b начиная с l . После основного цикла дописываются оба возможных остатка. Затем только $b[l...r-1]$ копируется обратно в a . Буфер создаётся один раз длиной всего вектора и передаётся по ссылке.

Время $O(n \log n)$ для целого вектора, буфер $O(n)$, стек $O(\log n)$. При равенстве $\leq$ выбирает левую половину, поддерживая устойчивость при сортировке записей по ключу. Контракт урока: до 100 целых; при прямом вызове ms требуется $0 \leq l <= r \leq a.size()$ и $b.size() = a.size()$. Границы в примерах заданы учителем и проверяются до вызова.

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 $[6, 2, 5, 1]$ вызов $[0, 4)$ делит в $[0, 2)$ и $[2, 4)$. После рекурсии $a = [2, 6, 1, 5]$. В буфер последовательно попадают 1, 2, 5, затем остаток 6. Копирование возвращает $[1, 2, 5, 6]$.

```
#include <iostream>
#include <vector>
void ms(std::vector<int>& a, std::vector<int>& b,
        std::size_t l, std::size_t r) {
    if(r-l<2) return;
    std::size_t m=l+(r-l)/2;
    ms(a,b,l,m); ms(a,b,m,r);
    std::size_t i=l, j=m, k=l;
    while(i<m && j<r) {
        if(a[i]<=a[j]) b[k++]=a[i++];
        else b[k++]=a[j++];
    }
    while(i<m) b[k++]=a[i++];
    while(j<r) b[k++]=a[j++];
    for(k=l; k<r; ++k) a[k]=b[k];
}
int main() {
    std::vector<int> a{6,2,5,1}, b(a.size());
    ms(a,b,0,a.size());
    for(int x:a) std::cout<<x<<' ';
    std::cout<<'\\n';
}
```



Границы	$[0,4) \rightarrow [0,2)$ и $[2,4)$
После рекурсии	2 6 1 5
Буфер $b[0:4]$	$1 \rightarrow 2 \rightarrow 5 \rightarrow 6$
Копирование назад	$a = [1,2,5,6]$

Правая граница не входит в участок; буфер имеет размер исходного вектора.

03. Практика 1. Проверяем материал

Сортируем весь вектор

14–25 минут. В `main` меняйте `a`, но оставляйте `b(a.size())` и правую границу `a.size()`.

Вход целиком	Ожидаемый итог
{6,2,5,1}	1 2 5 6
{ }; {8}	Пусто; 8
{3,-1,3,0,2}	-1 0 2 3 3
{1,2,3}; {3,2,1}	1 2 3 в обоих случаях

2 минуты — границы разбиений, 4 — шесть запусков, 3 — проверка длины и повторов, 2 — журнал. Критерии: нечётная длина обработана, пустота не обращается к элементам, оба остатка сохраняются.

04. Практика 2. Перенос способа

Сортируем только выбранный участок

25–33 минуты. Для $a=\{9,4,1,3,8\}$ и `b(a.size())` замените вызов на `ms(a,b,1,4)`. За 2 минуты прогноз, за 4 запуск и контроль внешних позиций, за 2 проверьте также `[2,2)` и `[0,5)` на свежих копиях. Критерии: три правильных результата и неизменность вне участка.

05. Обсуждение и границы вывода

33–40 минут. Один ученик создаёт буфер длиной $n-l$ и использует прежние индексы $k=l$. За 2 минуты найдите опасность на `[1,4)`, за 3 предложите две корректные схемы, за 2 выберите одну для этого урока.

06. Индивидуальная проверка

40–45 минут. Для участка `[2,7)` назовите длину, середину по формуле, дочерние участки и допустимость обращения `a[7]` внутри него. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для длины 5 первая середина равна 2: части длиной 2 и 3. Пустой [0,0) сразу завершает функцию. Если забыть хвост правой или левой части, в буфере останутся старые/нулевые значения, поэтому нужны тесты обоих направлений и с повторениями.

Практика 2: разбор и критерии

[1,4): [9,1,3,4,8]; [2,2): исходный [9,4,1,3,8]; [0,5): [1,3,4,8,9]. В первом случае позиции 0 и 4 остаются 9 и 8. Не заменяйте `r` на 3: это исключило бы элемент исходной позиции 3.

Обсуждение: возможный вывод

Буфер длиной 3 имеет индексы 0,1,2, но `k` достигает 3 — выход за границу. Можно сохранить буфер длиной `a.size()` и абсолютные индексы, как здесь, либо использовать относительные `k-l` и отдельную схему копирования. Нельзя смешивать две системы индексов.

Индивидуальная проверка: ответ и критерии

Длина 5; $m=2+(7-2)/2=4$; [2,4) и [4,7); `a[7]` не входит в участок. Фактическая допустимость `a[7]` во всём векторе зависит от его длины, но этот алгоритм читать его здесь не должен.

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`, рекурсия, полуинтервалы и ссылки. 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	Сортируем весь вектор	Реализовать слияние на полуинтервале $[l, r)$, проверить границы и сохранить элементы вне сортируемого участка.
Структурирование решения и отладка	Сортируем только выбранный участок	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: сортировка слиянием

<https://xlinux.nist.gov/dads/HTML/mergesort.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.