



Урок 29. C++: Инвариант и границы двоичного поиска

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 29 из 34 · 45 минут · Модуль «Поиск»

Уровень: начинающий программист; C++17.

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

Чтобы сосчитать повторы, не обязательно проходить весь диапазон: достаточно двух границ. Ошибка в одном знаке меняет смысл результата.

Результат урока: Найти позиции первого $\geq x$ и первого $> x$, вычислить количество вхождений и проверить инвариант.

01. Проблемная ситуация

0-5 минут. В $[-1, 2, 2, 2, 5]$ отметьте первую позицию, где значение не меньше 2, и первую, где строго больше 2. Почему разность позиций равна количеству двоек?

02. Понятия и разобранный пример

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

Две границы, два условия

`lower` возвращает число элементов $< x$, `upper` — число элементов $\leq x$. Поэтому равные x занимают полуинтервал $[lower, upper)$, а количество равно `upper - lower`. Обе функции возвращают позицию от 0 до n включительно, не обязательно индекс элемента.

В `lower` слева от l всё $< x$, начиная с r всё $\geq x$. В `upper` слева всё $\leq x$, справа всё $> x$. Неисследованная часть $[l, r)$ сокращается: ложная для правой части середина отбрасывается через $l = m + 1$, подходящая остаётся границей $r = m$. Установка $l = m$ при длине 1 зациклила бы поиск.

Контракт: отсортированный вектор не более 100 целых; n приводим к `int`. Пустой вектор имеет обе границы 0. Для подсчёта значений в закрытом диапазоне $[u, v]$ при $u \leq v$ используйте `upper(v) - lower(u)`, не обращаясь к `a[n]`.

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

`a = [-1, 2, 2, 2, 5]`, `x = 2`: `lower = 1`, `upper = 4`, количество 3. `lower`: $[0, 5) \rightarrow [0, 2) \rightarrow [0, 1) \rightarrow [1, 1)$. `upper`: $[0, 5) \rightarrow [3, 5) \rightarrow [3, 4) \rightarrow [4, 4)$.

```
#include <iostream>
#include <vector>
using namespace std;
int bound(const vector<int>& a, int x, bool upper) {
    int l=0, r=(int)a.size();
    while (l<r) {
        int m=l+(r-l)/2;
        bool left=upper ? a[m]<=x : a[m]<x;
        if (left) l=m+1;
        else r=m;
    }
    return l;
}
int main() {
    vector<int> a={-1,2,2,2,5};
    int lo=bound(a,2,false), hi=bound(a,2,true);
    cout << lo << ' ' << hi << ' ' << hi-lo;
}
```



Значения	-1 2 2 2 5
Индексы	0 1 2 3 4
Границы для 2	lower=1; upper=4
Количество	[1,4) содержит 4-1=3 элемента

Правую границу не включают в диапазон равных значений.

03. Практика 1. Проверяем материал

Проверяем интервалы равных

14–25 минут. Учебные векторы уже отсортированы. Рисуйте границы между клетками, а не поверх чисел.

Вектор; x	Найдите lower, upper, количество
[-1,2,2,2,5]; 2	Повторяющееся значение
[-1,2,2,2,5]; 3	Между значениями
[4,4]; 4	Все равны
[]; 0	Пустота
[1]; 0 и 2	До начала и после конца

2 минуты — прогноз, 4 — шесть запусков, 3 — проверка счётом вручную, 2 — журнал границ. Критерии: позиции в 0...n, неотрицательная разность, совпадение с ручным счётом и корректная пустота.

04. Практика 2. Перенос способа

Ищем значения в диапазоне

25–33 минуты. Для [-3,-1,-1,0,4,4,7] сосчитайте элементы закрытых диапазонов [-1,4], [1,3], [8,9]. За 2 минуты определите нужные вызовы, за 4 проверьте программно, за 2 объясните, почему нельзя считать lower(v)-lower(u).

05. Обсуждение и границы вывода

33–40 минут. В ошибочной версии стоит l=m. За 2 минуты возьмите [1],x=2, за 3 проследите повторение состояния, за 2 сформулируйте условие прогресса цикла.

06. Индивидуальная проверка

40–45 минут. Для [1,3,3,8] и x=3 запишите обе границы, количество и безопасное правило проверки наличия. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Тройки ответов: (1,4,3), (4,4,0), (0,2,2), (0,0,0), для 1: (0,0,0), для 2: (1,1,0). Отсутствие числа не означает границу -1: обе позиции вставки могут совпасть внутри или на краю.

Практика 2: разбор и критерии

Для [-1,4]: $\text{upper}(4)=6$, $\text{lower}(-1)=1$, ответ 5. [1,3]: $4-4=0$. [8,9]: $7-7=0$. $\text{lower}(v)$ исключил бы значения, равные правой границе v , поэтому для первого диапазона потерялись бы две четвёрки. При $u > v$ сначала отвергают некорректный диапазон или отдельно определяют контракт.

Обсуждение: возможный вывод

$l=0, r=1, m=0$; $a[m] < 2$, но $l=m$ оставляет $l=0$. Следующий шаг идентичен. Правильное $l=m+1$ даёт $l=r=1$. Инвариант без уменьшения $r-l$ недостаточен: нужно отдельно доказать завершение.

Индивидуальная проверка: ответ и критерии

$\text{lower}=1$, $\text{upper}=3$, количество 2. Наличие можно проверить $\text{upper} > \text{lower}$ либо $\text{lower} < n \ \&\& \ a[\text{lower}] = x$. По баллу за каждый пункт.

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: полуинтервал $[l, r)$, vector и двоичный поиск первого вхождения. 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 поиска	Проверяем интервалы равных	Найти позиции первого $\geq x$ и первого $> x$, вычислить количество вхождений и проверить инвариант.
Самостоятельная проверка результата	Ищем значения в диапазоне	Обоснованный ответ и исправление ошибки

Источники и права на изображения

Исходная программа курса

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

Проект стандарта C++: lower_bound и upper_bound

<https://eel.is/c++draft/alg.binary.search>

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.