



Урок 28. C++: Линейный и двоичный поиск

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 28 из 34 · 45 минут · Модуль «Поиск»

Уровень: начинающий программист; C++17.

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

Линейный поиск можно начать в любом списке. Двоичный сокращает область поиска, но ошибается, если нарушена упорядоченность.

Результат урока: Реализовать два поиска первого вхождения и выбрать алгоритм с учётом порядка исходных данных.

01. Проблемная ситуация

0–5 минут. В [8,2,6,2] найдите первую двойку слева. Затем упорядочьте значения и снова найдите двойку. Совпали ли индексы? Какое значение требуется вернуть, если числа нет?

02. Понятия и разобранный пример

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

Контракт индекса и порядка

Индексы нумеруются с нуля. Линейная функция просматривает элементы слева и возвращает первый подходящий индекс, иначе -1. Она работает и на неупорядоченных данных. Учебный размер от 0 до 100 элементов делает преобразование `size()` в `int` безопасным.

Двоичная функция ищет левую границу в отсортированном по неубыванию векторе. За пределами $[l, r)$ слева элементы меньше x , справа не меньше x . Когда $l=r$, это позиция вставки; для ответа о наличии дополнительно проверяем $l < n$ и $a[l] == x$. Порядок условий $\&\&$ важен: при $l = n$ нельзя читать $a[l]$.

Линейный поиск в худшем случае делает n сравнений равенства; двоичный с доступом по индексу — порядка $\log_2(n)$ шагов. Если ради одного запроса сначала сортировать, подготовка тоже стоит времени и меняет индексы. Полный выбор зависит от числа запросов и нужного смысла индекса.

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

В отсортированном $[-2, 1, 1, 4, 9]$ для $x=1$: $l=0, r=5, m=2 \rightarrow r=2$; $m=1 \rightarrow r=1$; $m=0 \rightarrow l=1$. Позиция 1, значение равно x , ответ 1. Для $x=10$ граница 5, значит ответа нет.

```
#include <iostream>
#include <vector>
using namespace std;
int linear(const vector<int>& a, int x) {
    for (int i=0; i<(int)a.size(); ++i)
        if (a[i]==x) return i;
    return -1;
}
int binary_first(const vector<int>& a, int x) {
    int l=0, r=(int)a.size();
    while (l<r) {
        int m=l+(r-l)/2;
        if (a[m]<x) l=m+1;
        else r=m;
    }
    if (l<(int)a.size() && a[l]==x) return l;
    return -1;
}
int main() {
    vector<int> a={-2,1,1,4,9};
    cout << linear(a,1) << ' ' << binary_first(a,1);
}
```



Область [l,r)	$[0,5) \rightarrow [0,2) \rightarrow [0,1) \rightarrow [1,1)$
Середина m	$2 \rightarrow 1 \rightarrow 0$
Проверка	<code>a[1] == 1</code>
Ответ	Первое вхождение: индекс 1

Левая граница позволяет найти первое из нескольких равных значений.

03. Практика 1. Проверяем материал

Сравниваем два результата

14–25 минут. Каждый тест запустите обеими функциями на одном и том же векторе.

Вектор; x	Случай
<code>[-2,1,1,4,9]; 1</code>	Повтор
<code>[-2,1,1,4,9]; -2, 9, 10</code>	Края и отсутствие
<code>[]; 0</code>	Пустой вектор
<code>[5]; 5, 4</code>	Один элемент

2 минуты — ожидаемые индексы; 4 — семь пар запусков; 3 — трасса для `x=10`; 2 — журнал «вектор / `x` / `linear` / `binary_first`». Критерии: оба результата совпали, первый повтор выбран, пустой ввод безопасен.

04. Практика 2. Перенос способа

Сохраняем исходное положение

25–33 минуты. Вектор `[8,2,6,2]` не отсортирован, требуется исходный индекс первой двойки. За 2 минуты выполните `linear`, за 3 проследите `binary_first` на неверном входе, за 3 предложите решение для одного запроса и для многих запросов с сохранением исходных позиций. Код двоичного поиска менять не требуется.

05. Обсуждение и границы вывода

33–40 минут. Один разработчик обещает ускорение любого поиска, другой предлагает учесть подготовку. За 2 минуты назовите цену сортировки, за 3 сравните один и миллион запросов к неизменному вектору, за 2 запишите обоснованный выбор.

06. Индивидуальная проверка

40–45 минут. В `[0,0,3,7]` найдите первый 0 и ответ для 8. Укажите предусловие и объясните проверку `l < p`. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Индексы по порядку: 1; 0,4,-1; -1; 0,-1. Для 10 границы [0,5)→[3,5)→[5,5); проверка $l < n$ ложна. Не следует возвращать просто l : позиция вставки 5 не является индексом найденного элемента.

Практика 2: разбор и критерии

linear возвращает 1. Неверный запуск binary_first: $m=2,6 \geq 2 \rightarrow r=2$; $m=1,2 \geq 2 \rightarrow r=1$; $m=0,8 \geq 2 \rightarrow r=0$; $a[0] \neq 2 \rightarrow -1$. Для одного запроса достаточно линейного прохода. Для многих можно подготовить пары (значение,исходный индекс) и сортировать по значению, затем индексу: (2,1),(2,3),(6,2),(8,0); первый элемент диапазона двоек хранит исходный индекс 1.

Обсуждение: возможный вывод

Уже отсортированный вектор хорошо подходит двоичному поиску. Сортировка ради одного запроса может стоить больше линейного прохода. Для многих запросов подготовка используется повторно. При изменениях данных порядок надо поддерживать; индекс после сортировки не равен исходному.

Индивидуальная проверка: ответ и критерии

Для 0 ответ 0, для 8 — -1. Предусловие: неубывающий порядок. $l < n$ защищает от обращения за концом, в том числе при пустом векторе. 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: vector, индексы, циклы, возврат значения функции. 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	Сравниваем два результата	Реализовать два поиска первого вхождения и выбрать алгоритм с учётом порядка исходных данных.
Самостоятельная проверка результата	Сохраняем исходное положение	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++: условия двоичного поиска

<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.