



Урок 26. Python: Инвариант и границы двоичного поиска

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 26 из 34 · 45 минут · Модуль «Поиск»

Уровень: начинающий программист; Python 3.

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

Поиск границы отвечает не только «есть ли число». Две границы позволяют сосчитать все повторения без просмотра каждого совпадения.

Результат урока: Реализовать левую и правую границы, доказать сохранение инварианта и вычислить число значений в заданном диапазоне.

01. Проблемная ситуация

0–5 минут. В [1,2,2,2,5,7] найдите позицию перед первой 2 и после последней 2. Между какими индексами находятся совпадения? Сколько их и почему нельзя просто вычесть индексы первого и последнего?

02. Понятия и разобранный пример

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

Две границы и доказательство

lower ищет первый индекс с $a[i] \geq x$, upper — первый с $a[i] > x$. Если такой позиции нет, ответ n. Это позиции вставки, а не обязательно индексы существующих элементов. Для lower все позиции левее lo имеют значения $< x$, все позиции от hi до конца имеют $\geq x$. Для upper знаки меняются на $\leq x$ и $> x$.

Середина находится внутри [lo,hi). Если проверка разрешает отбросить левую часть, $lo = mid + 1$; иначе $hi = mid$. Неизвестный интервал строго сокращается, поэтому цикл заканчивается. При $lo = hi$ все позиции разделены нужным условием. Присваивание $lo = mid$ может застрять на интервале длины 1.

Повторы x занимают полуинтервал $[lower(a,x), upper(a,x))$, количество равно разности границ. Для закрытого диапазона $L \leq value \leq R$ при $L \leq R$ количество $upper(a,R) - lower(a,L)$. Каждый поиск $O(\log n)$, дополнительная память $O(1)$. Предпосылка: отсортированный список до 100 целых; пустота разрешена.

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 [1,2,2,2,5,7] и $x=2$ lower даёт 1, upper даёт 4: совпадения на индексах 1,2,3, их $4-1=3$. Если $x=3$, обе границы 4 и число совпадений 0. Для $x=9$ обе границы 6; индекс 6 не читаем, это конец.

```
def lower(a, x):
    lo, hi = 0, len(a)
    while lo < hi:
        mid = (lo + hi) // 2
        if a[mid] < x:
            lo = mid + 1
        else:
            hi = mid
    return lo

def upper(a, x):
    lo, hi = 0, len(a)
    while lo < hi:
        mid = (lo + hi) // 2
        if a[mid] <= x:
            lo = mid + 1
        else:
            hi = mid
    return lo

a = [1, 2, 2, 2, 5, 7]
for x in [2, 3, 9]:
    l, r = lower(a,x), upper(a,x)
    print(x, l, r, r-l)
```



Значения	1 2 2 2 5 7
Индексы совпадений	1,2,3
Границы для $x=2$	lower=1; upper=4
Кратность	Длина [1,4) = 4-1 = 3

Правая граница расположена после последнего совпадения.

03. Практика 1. Проверяем материал

Считаем кратность значения

14–25 минут. Для каждого запроса выпишите границы и разность; значения $a[upper]$ читать не требуется.

Вход; x	lower, upper, количество
[1,2,2,2,5,7];2	1,4,3
Тот же ряд;3;9	4,4,0; 6,6,0
[];4	0,0,0
[4,4];4;3;5	0,2,2; 0,0,0; 2,2,0

2 минуты — прогноз, 4 — семь запросов, 3 — трасса upper для $x=2$, 2 — объяснение инварианта. Критерии: все границы допустимы от 0 до n , длина интервала совпадает с ручным подсчётом, пустота обработана.

04. Практика 2. Перенос способа

Считаем значения в диапазоне

25–33 минуты. Используйте тот же a и найдите количество значений в закрытых диапазонах [2,5], [3,4], [0,1], [8,9]. За 2 минуты получите формулу, за 4 выполните четыре расчёта функциями, за 2 решите, что делать при $L>R$. Форма: L, R / левая граница / правая граница / число. Критерии: четыре ответа и явное правило неверного порядка.

05. Обсуждение и границы вывода

33–40 минут. Почему проверка $a[mid]==x$ с немедленным return не подходит для подсчёта повторов? За 2 минуты разберите основной ряд, за 3 сформулируйте новый контракт, за 2 объясните, какие тесты отличат реализации.

06. Индивидуальная проверка

40–45 минут. Для [0,0,3,6,6] найдите lower(6), upper(6), число значений в [0,3] и объясните зависание при $lo=mid$ на $[lo,hi)=(4,5)$, если условие требует идти вправо. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

upper для $x=2$: $[0,6)$, $mid=3$, $a[3]=2 \Rightarrow lo=4$; $[4,6)$, $mid=5$, $7>2 \Rightarrow hi=5$; $[4,5)$, $mid=4$, $5>2 \Rightarrow hi=4$. Ответ 4. Слева от границы значения ≤ 2 , справа >2 . Для отсутствующих значений границы равны; это не ошибка.

Практика 2: разбор и критерии

Для $[2,5]$: $lower(2)=1$, $upper(5)=5$, число 4. $[3,4]:4,4,0$. $[0,1]:0,1,1$. $[8,9]:6,6,0$. При $L>R$ объявляем ошибку входа, например `raise ValueError("L > R")`, до вычитания. Нельзя выдавать отрицательное число за количество; допустима другая явно заданная политика, но не молчаливая перестановка границ.

Обсуждение: возможный вывод

Обычный поиск может вернуть индекс 3, хотя первая двойка стоит на 1; единственное найденное совпадение не даёт кратности. Для подсчёта нужны именно две границы. Тесты: все элементы равны, повтор в начале и конце, отсутствующий x между значениями и за обоими краями, пустой список.

Индивидуальная проверка: ответ и критерии

$lower(6)=3$; $upper(6)=5$; $upper(3)-lower(0)=3-0=3$; $mid=4$, поэтому $lo=mid$ оставляет $[4,5)$ неизменным, тогда как $lo=mid+1$ завершает поиск. По одному баллу.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. 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: поиск левой границы из урока 25, полуинтервалы и доказательство завершения. 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	Считаем кратность значения	Реализовать левую и правую границы, доказать сохранение инварианта и вычислить число значений в заданном диапазоне.
Структурирование решения и отладка	Считаем значения в диапазоне	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>

Python: `bisect_left` и `bisect_right`

<https://docs.python.org/3/library/bisect.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.