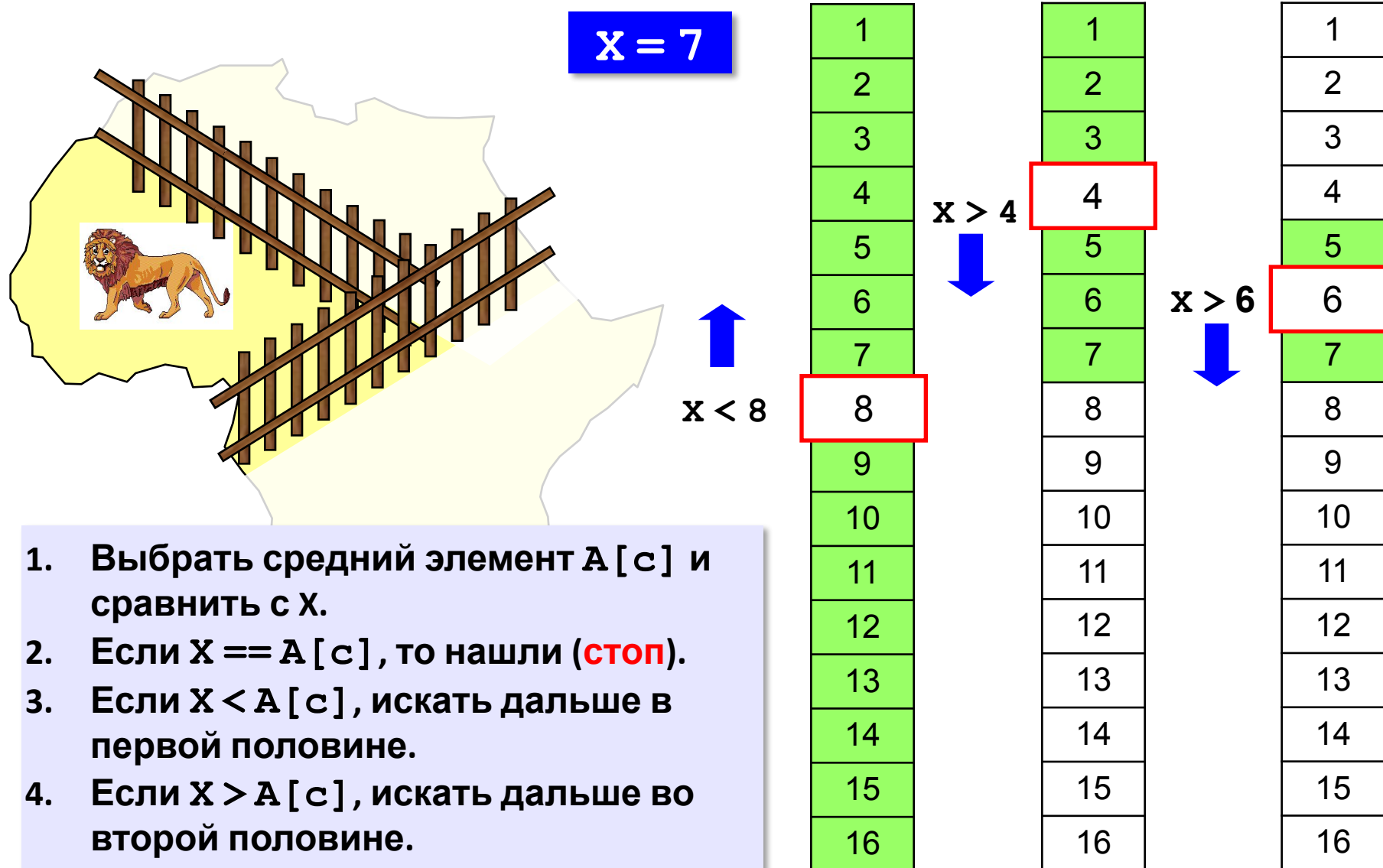


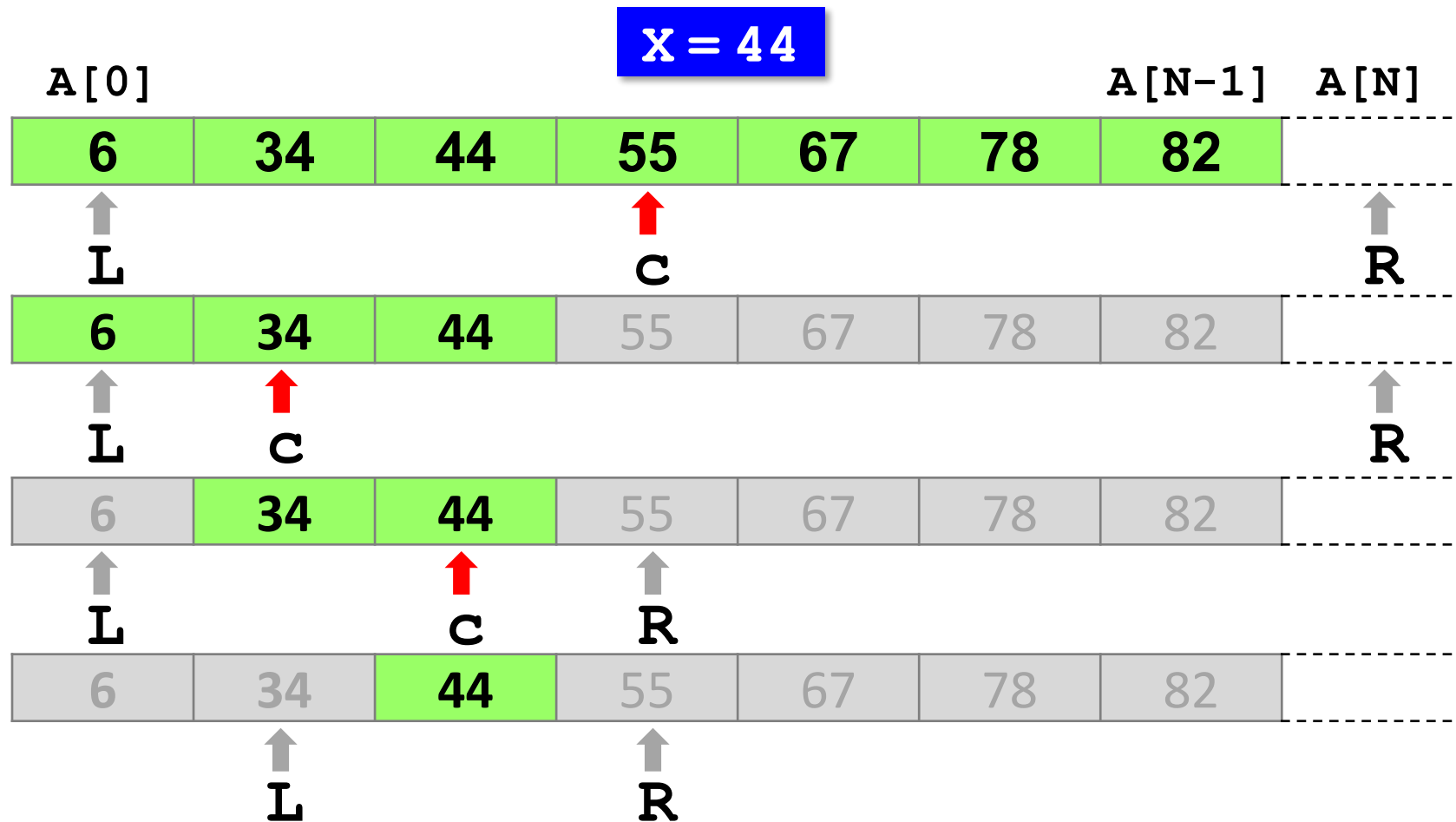
Программирование на языке Python

§ 65. Двоичный поиск

Двоичный поиск



Двоичный поиск



L = R - 1 : поиск завершен!



Двоичный поиск

```
L = 0; R = N      # начальный отрезок
while L < R - 1:
    c = (L + R) // 2 # нашли середину
    if X < A[c]:     # сжатие отрезка
        R = c
    else: L = c
if A[L] == X:
    print ( "A[" , L, "]" = " , X, sep = "" )
else:
    print ( "Не нашли!" )
```

Двоичный поиск

Число сравнений:

N	линейный поиск	двоичный поиск
2	2	2
16	16	5
1024	1024	11
1048576	1048576	21

-  скорость выше, чем при линейном поиске
-  нужна предварительная сортировка

 Когда нужно применять?

Задачи

«А»: Заполнить массив случайными числами и отсортировать его. Ввести число X . Используя двоичный поиск, определить, есть ли в массиве число, равное X . Подсчитать количество сравнений.

Пример:

Массив :

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X :

2

Число 2 найдено.

Количество сравнений: 2

Задачи

«В»: Заполнить массив случайными числами и отсортировать его. Ввести число X. Используя двоичный поиск, определить, сколько чисел, равных X, находится в массиве.

Пример:

Массив :

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X:

4

Число 4 встречается 2 раз (а) .

Пример:

Массив :

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 7 9

Введите число X:

14

Число 14 не встречается.

Задачи

«С»: Заполнить массив случайными числами и ввести число и отсортировать его. Ввести число X . Используя двоичный поиск, определить, есть ли в массиве число, равное X . Если такого числа нет, вывести число, ближайшее к X .

Пример:

Массив :

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 12 19

Введите число X :

12

Число 12 найдено.

Пример:

Массив :

1 4 7 3 9 2 4 5 2

После сортировки:

1 2 2 3 4 4 5 12 19

Введите число X :

11

Число 11 не найдено. Ближайшее число 12.