АКАДЕМИЯ АЙТИ

День 10

academyit.ru

Типовые задачи на обработку текста.

Практические примеры составления блоксхем и псевдокода.
Простейшие алгоритмические задачи.
Перевод алгоритма в код.
Подпрограммы (функции) как основные блоки кода.

Турашова Анна Николаевна
Преподаватель
anna1turashova@gmail.com
Telegram: @anna1tur

Поверка домашнего задания

Задача №3752. Встречалось ли число раньше

Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово `YES` (в отдельной строке), если это число ранее встречалось в последовательности или `NO`, если не встречалось.

Входные данные

Вводится список чисел. Все числа списка находятся на одной строке.

Выходные данные

Выведите ответ на задачу.

Примеры

входные данные
1 2 3 2 3 4
выходные данные
NO NO NO YES YES NO

Задача №112364. Поиск в матрице

Напишите программу, которая определяет, сколько раз встречается в матрице элемент, равный K.

Входные данные

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \le N, M \le 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами. В следующей строке записано целое число K.

Выходные данные

Программа должна вывести количество элементов матрицы, равных K.

Примеры

входные данные
4 5 1 2 3 4 5 6 12 8 9 10 11 12 12 14 15 16 17 18 12 20 12
выходные данные
4

Задача №112368. Столбцы с максимумом

Напишите программу, которая находит в матрице столбцы, в которых есть элемент, равный максимальному.

Входные данные

В первой строке записаны через пробел размеры матрицы: количество строк N и количество столбцов M ($1 \le N, M \le 100$). В следующих N строках записаны строки матрицы, в каждой – по M натуральных чисел, разделённых пробелами.

Выходные данные

Программа должна вывести все столбцы, в которых есть элемент, равный максимальному элементу в матрице. Каждый столбец выводится в одну строку, элементы разделяются пробелами.

Примеры

входные данные
4 5 1 897 2 54 234 75 12 3 46 9 13 26 56 9 12 14 90 897 6 34
выходные данные
897 12 26 90 2 3 56 897

Задача №354. Побочная диагональ

Дано число `n`, `n \(\leq\) 100` . Создайте массив `n×n` и заполните его по следующему правилу:
- числа на диагонали, идущей из правого верхнего в левый нижний угол, равны 1;
- числа, стоящие выше этой диагонали, равны 0;
- числа, стоящие ниже этой диагонали, равны 2.

Входные данные

Программа получает на вход число `n`.

Выходные данные

Необходимо вывести полученный массив. Числа разделяйте одним пробелом.

Примеры

входные данные
4
выходные данные
0 0 0 1 0 0 1 2 0 1 2 2 1 2 2 2

Задача №355. Симметричная ли матрица?

Проверьте, является ли двумерный массив симметричным относительно главной диагонали. Главная диагональ — та, которая идёт из левого верхнего угла двумерного массива в правый нижний.

Входные данные

Программа получает на вход число `n \(\leq\) 100`, являющееся числом строк и столбцов в массиве. Далее во входном потоке идет `n` строк по `n` чисел, являющихся элементами массива.

Выходные данные

Программа должна выводить слово `yes` для симметричного массива и слово `no` для несимметричного.

Примеры

входные данные
3 0 1 2 1 5 3 2 3 4
выходные данные
yes

Повторение:
Матрица

Создание матриц

Матрица – это список списков

```
a = [[-1, 0, 1],
     [-1, 0, 1],
     [0, 1, -1]]
```

```
a = [[-1, 0, 1], [-1, 0, 1], [0, 1, -1]]
```

Каждый элемент имеет два индекса, нумерация элементов с нуля

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]
a[2][0]	a[2][1]	a[2][2]

Ввод элементов с клавиатуры

Каждая строка таблицы на отдельной строке, значения в строке разделяются пробелами)

```
table = []
for i in range(n):
   row = [int(x) for x in input().split()]
   table.append(row)

table = [[int(x) for x in input().split()]
     for i in range(n)]
```

Вывод элементов в консоль

```
for row in a:
    for x in row:
      print (x, end = "\t" )
    print()


for i in range(len(a)):
    for j in range(len(a[i])):
      print (f"{a[i][j]:4d}", end = "")
    print()
```

Выделение строк, столбцов

Выделение первой строки

```
r = a[1][:]
```

Выделение третьего столбца

```
c = [row[3] for row in a]
```

Выделение главной диагонали

```
d = [a[i][i] for i in range(len(a))]
```

Алгоритмы сортировки. Алгоритмы поиска.

Задача №4. Двоичный поиск

Реализуйте алгоритм бинарного поиска.

Входные данные

В первой строке входных данных содержатся натуральные числа N и K ($0 < N, K \leq 100\,000$). Во второй строке задаются N элементов первого массива, отсортированного по возрастанию, а в третьей строке – K элементов второго массива. Элементы обоих массивов - целые числа, каждое из которых по модулю не превосходит 10^9

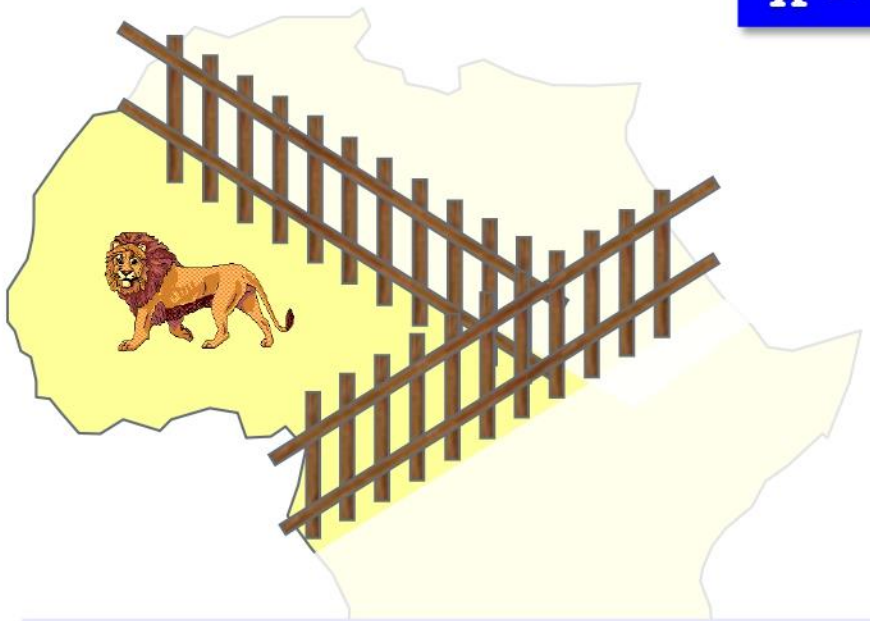
Выходные данные

Требуется для каждого из *K* чисел вывести в отдельную строку "YES", если это число встречается в первом массиве, и "NO" в противном случае.

Примеры

входные данные
10 5 1 2 3 4 5 6 7 8 9 10 -2 0 4 9 12
выходные данные
NO NO YES YES NO

Двоичный поиск



x = 7

x < 8

x > 4

x > 6

1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16

1. Выбрать средний элемент `a[c]` и сравнить с x.
2. Если `x == a[c]`, то нашли (**стоп**).
3. Если `x < a[c]`, искать дальше в первой половине.
4. Если `x > a[c]`, искать дальше во второй половине.

Алгоритм двоичного поиска

```
L,R = 0, 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(f"a[{L}]={x}")
else:
   print("Не нашли!")
```

Хранение элементов множества в неупорядоченном списке

Множество – структура данных, которая реализует хранение элементов без повторений и операции поиска, добавления, удаления, а также поиска минимального/максимального элемента

Поиск элемента (in, index)	O(n)
Добавление элемента (append)	O(1)
Удаление элемента (del)	O(n)
Поиск минимального/максимального элемента	O(n)

Хранение элементов множества в упорядоченном списке

Поиск элемента (двоичный поиск)	O(log n)
Добавление элемента (вставка)	O(n)
Удаление элемента (del)	O(n)
Поиск минимального/максимального элемента	O(1)

Сортировка

Сортировка – это расстановка элементов массива в заданном порядке.

Простые неэффективные алгоритмы – в худшем случае сложность $O(n^2)$:

- сортировка пузырьком
- сортировка вставками

Сложные эффективные алгоритмы – в худшем случае сложность $O(n \log n)$:

- «быстрая сортировка» (QuickSort)
- пирамидальная сортировка (HeapSort)
- сортировка слиянием (MergeSort)

Задача №233. Пузырьковая сортировка_0

Требуется отсортировать массив по неубыванию методом "пузырька".

Входные данные

В первой строке вводится одно натуральное число, не превосходящее 1000 – размер массива. Во второй строке задаются *N* чисел – элементы массива (целые числа, не превосходящие по модулю 1000).

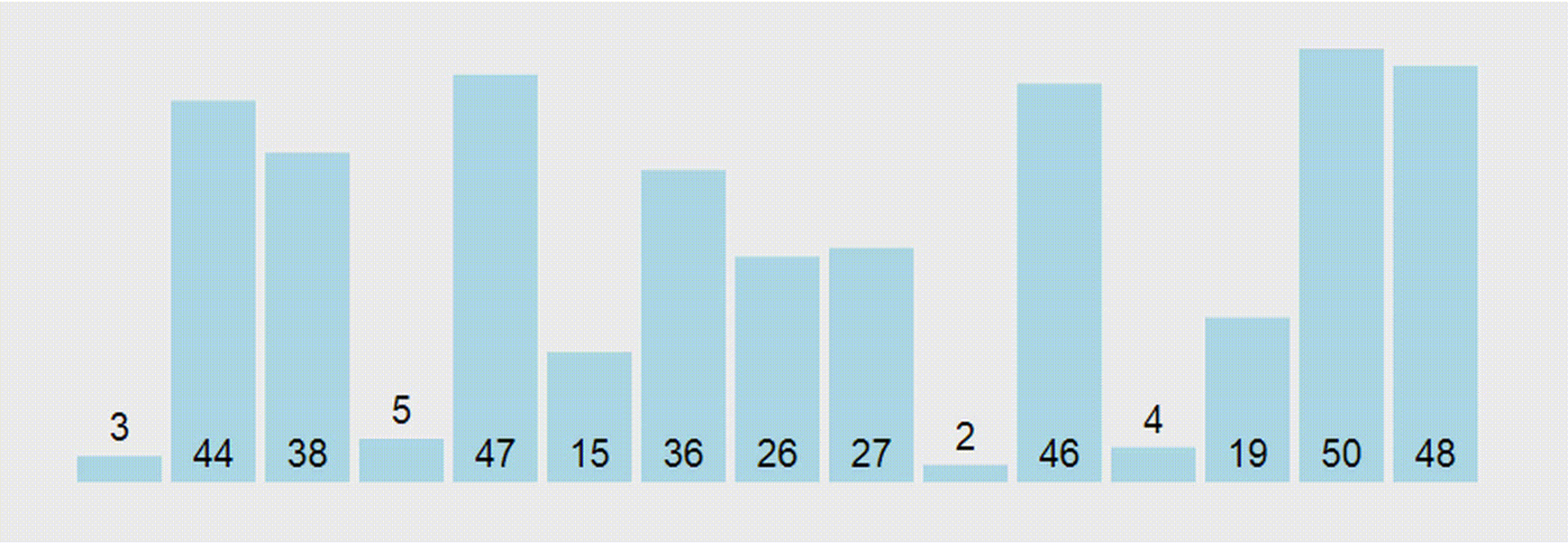
Выходные данные

Вывести получившийся массив.

Примеры

входные данные
5 5 4 3 2 1
выходные данные
1 2 3 4 5

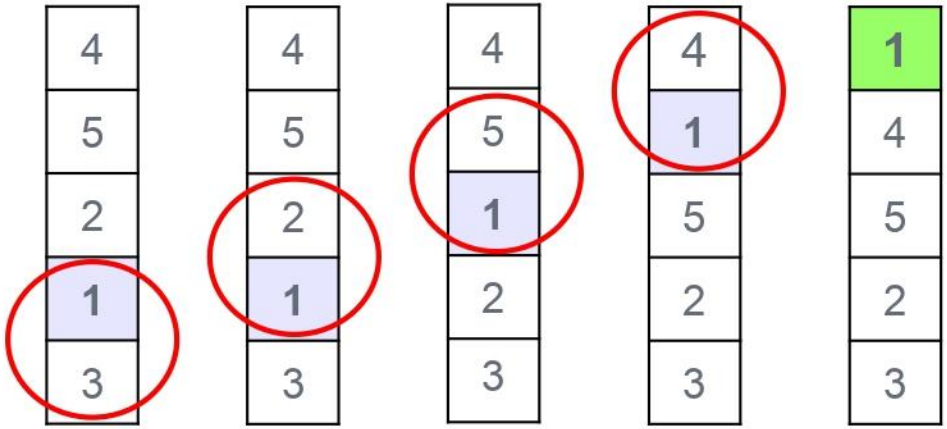
Bubble sort - сортировка пузырьком



Метод пузырька

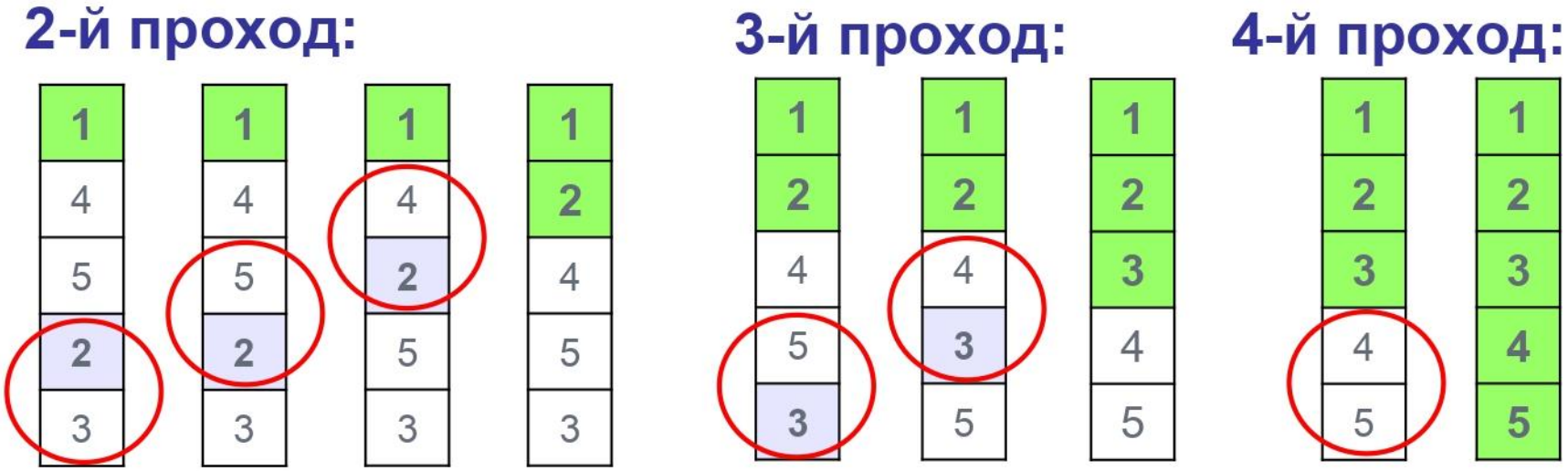
Идея алгоритма – самый маленький элемент перемещается в начало списка (всплывает как пузырек)

1-й проход:



- сравниваем два соседних элемента; если они стоят «неправильно», меняем их местами
- за 1 проход по массиву один элемент (самый маленький) становится на свое место

Метод пузырька



Для сортировки нужен n-1 проход.

```
for i in range(n-1):
  for j in range(n-2, i-1 ,-1):
    if a[j+1] < a[j]:
      a[j], a[j+1] = a[j+1], a[j]
```

6 5 3 1 8 7 2 4

Задача №1436. Библиотечный метод

Продемонстрируйте работу метода сортировки вставками по возрастанию. Для этого выведите состояние данного массива после каждой вставки на отдельных строках. Если массив упорядочен изначально, то следует не выводить ничего.

Входные данные

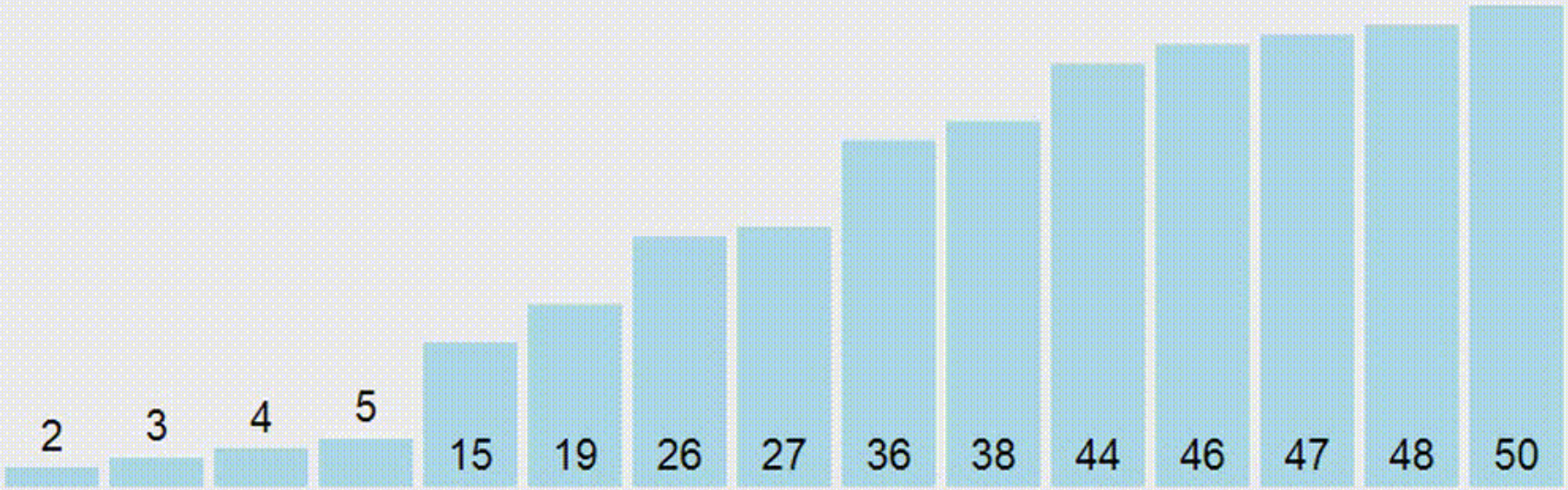
На первой строке дано число ($1 \le N \le 100$) – количество элементов в массиве. На второй строке задан сам массив: последовательность натуральных чисел, не превышающих 10^9.

Выходные данные

В выходной файл выведите строки (по количеству вставок) по N чисел каждая.

входные данные
4 2 1 5 3
выходные данные
1 2 5 3 1 2 3 5

Insertion sort - сортировка вставками



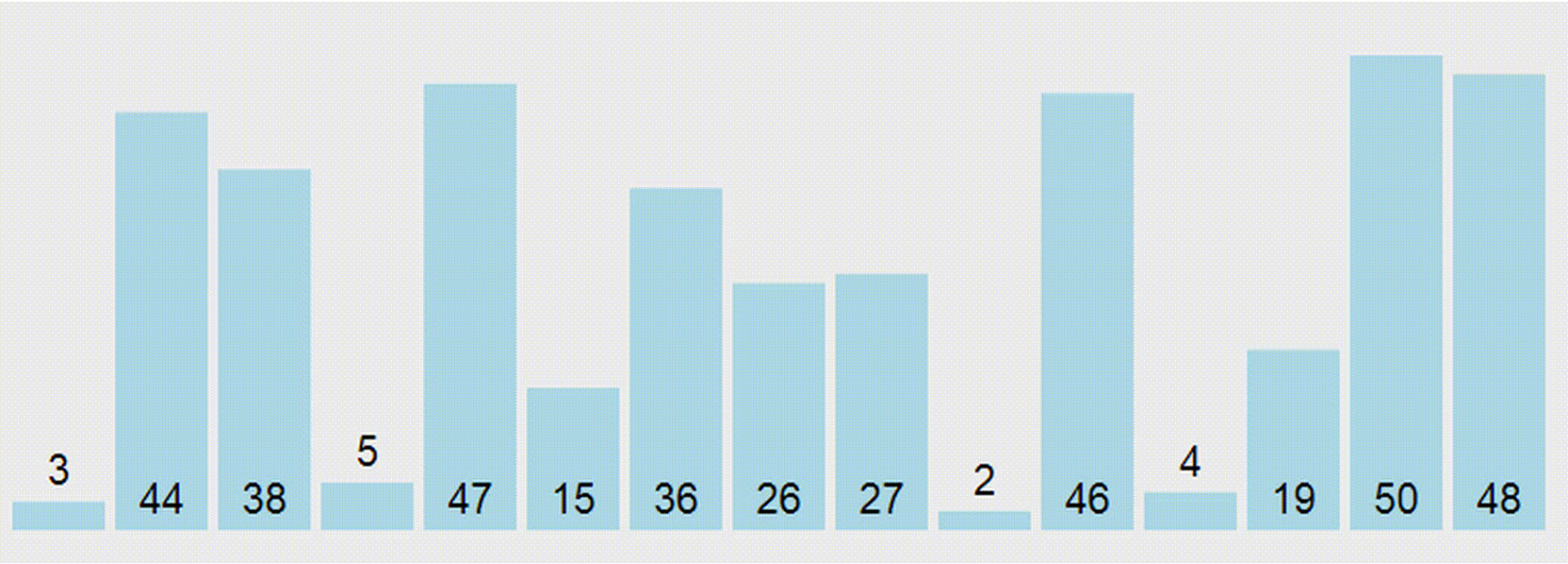
Метод вставок

Идея алгоритма – на каждом шаге *i* элемент вставляется на свое место среди предыдущих элементов.

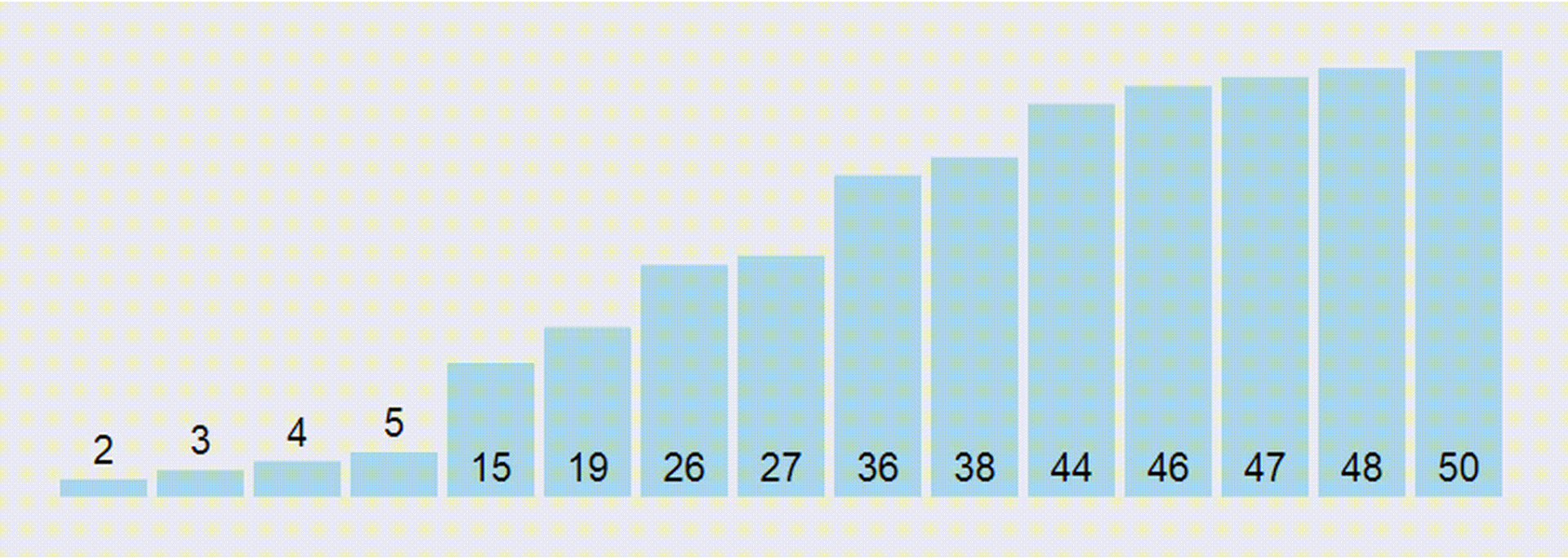
6 5 3 1 8 7 2 4

```
# Сортировку начинаем со второго элемента,
for i in range(1, n):
    item_to_insert = a[i]
    # Сохраняем ссылку на индекс предыдущего элемента
    j = i - 1
    # Элементы перемещаем вперёд, если они больше
    # элемента для вставки
    while j >= 0 and a[j] > item_to_insert:
        a[j + 1] = a[j]
        j -= 1
    # Вставляем элемент
    a[j + 1] = item_to_insert
```

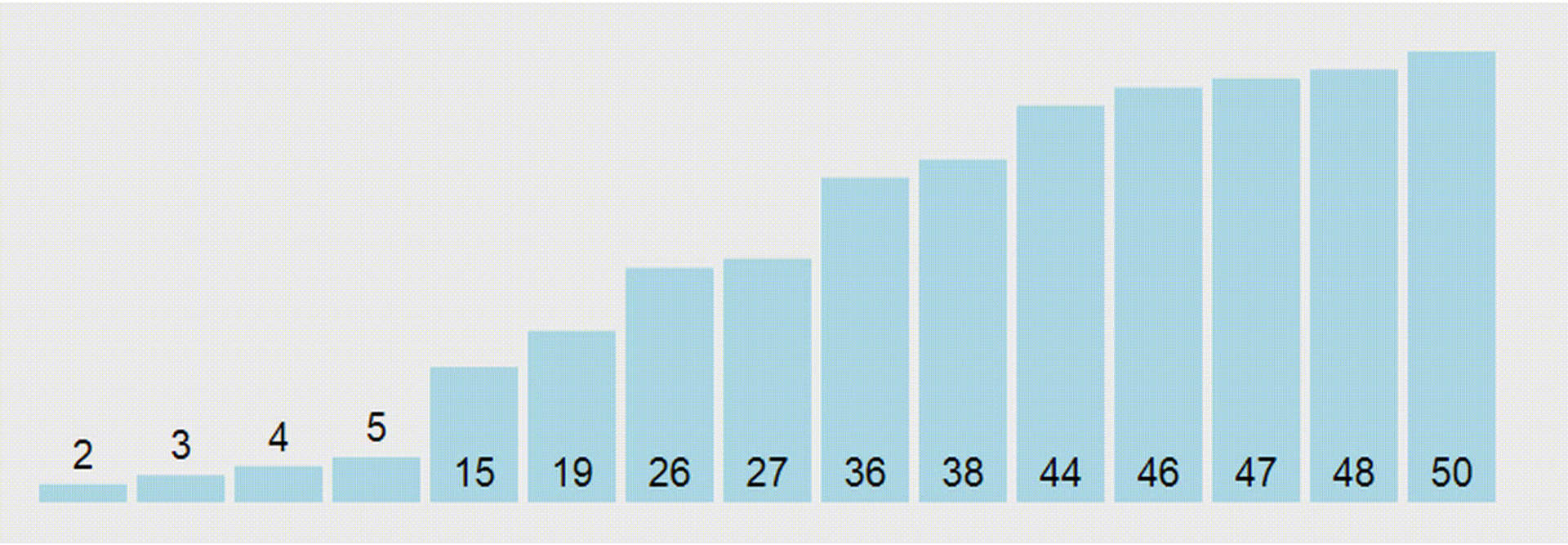
Selection sort - сортировка выбором



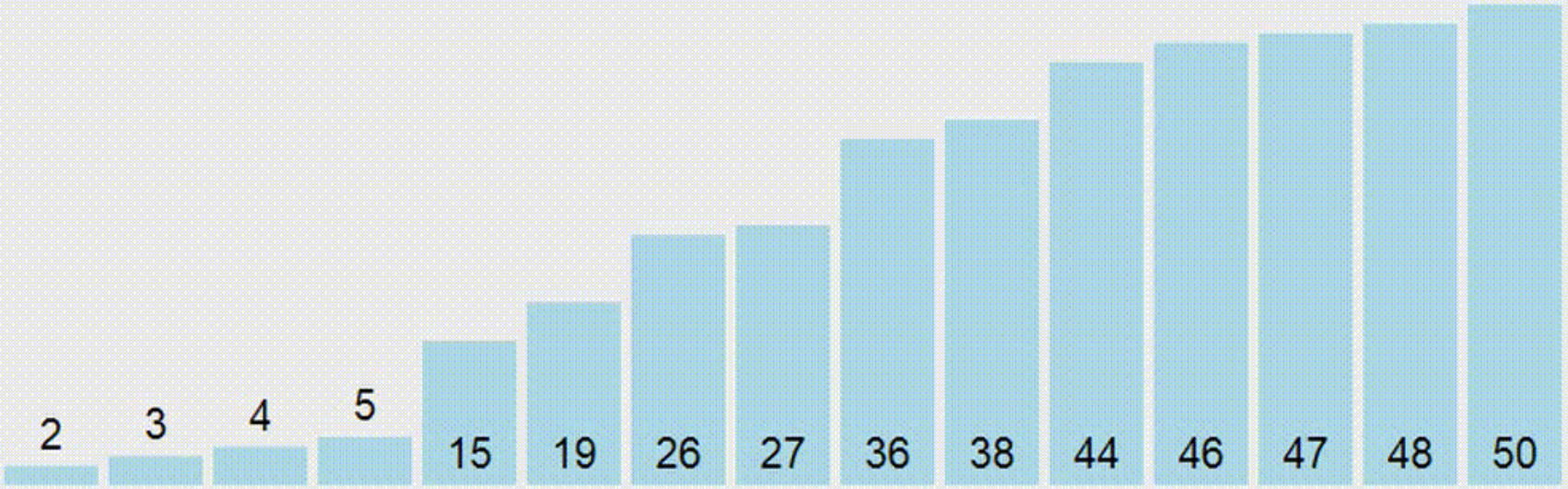
Quick sort - быстрая сортировка



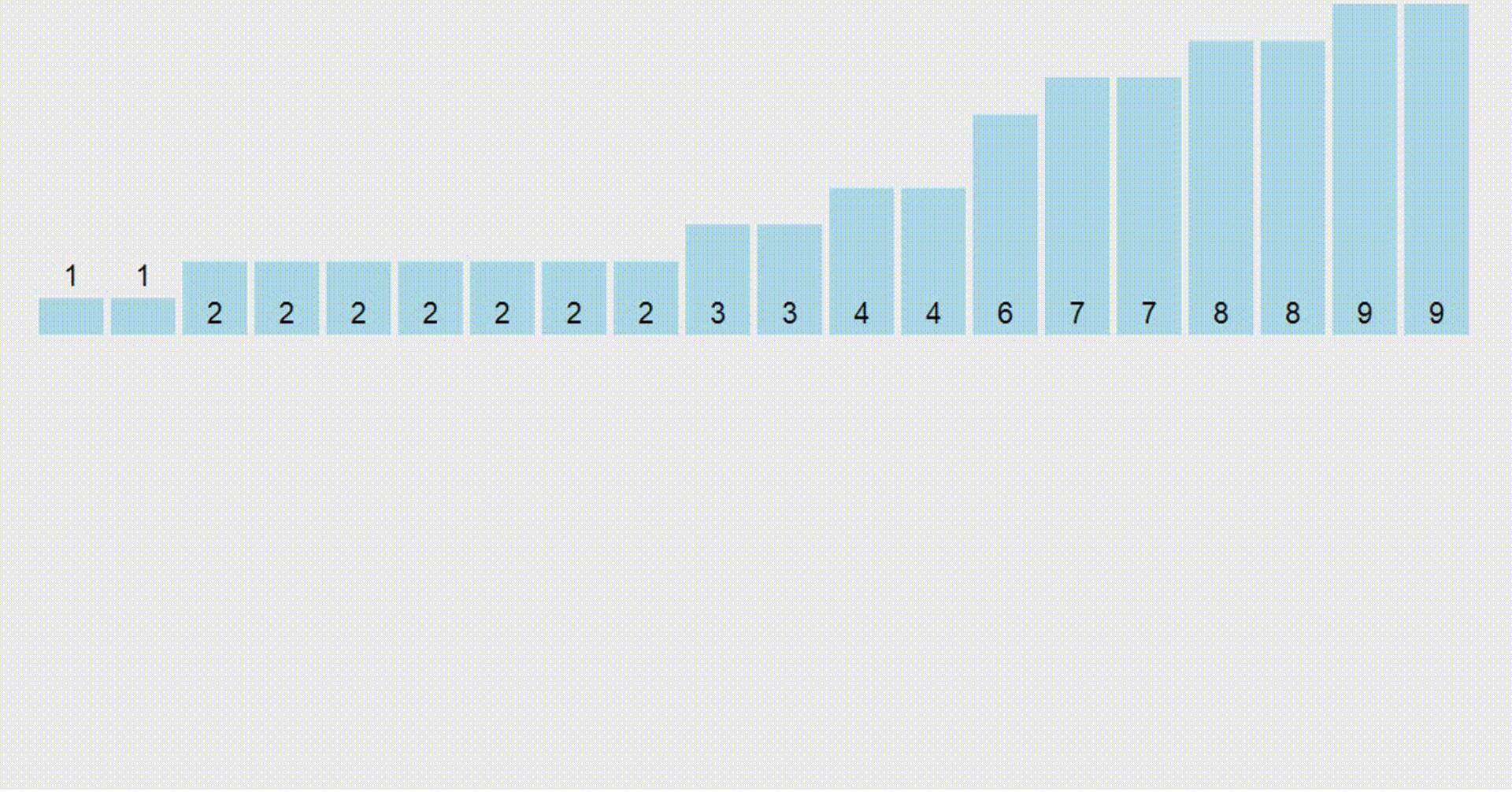
Random quick sort – рандомная быстрая сортировка



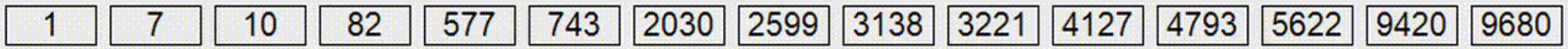
Merge sort – сортировка слиянием



Counting sort – сортировка подсчетом



Radix sort – поразрядная сортировка



Сравнение сортировок

Раз	Пузырьковая	Выборкой	Вставками	Куча	Слиянием	Быстрая
1	5.5318861007	1.2315289974	1.6035542488	0.0400667190	0.0261991024	0.0163919925
2	4.9217622280	1.2472858428	1.5910329818	0.0399959087	0.0258429050	0.0166139602
3	4.9164218902	1.2244019508	1.5936298370	0.0440728664	0.0286228656	0.0164628028
4	5.1547043323	1.2505383491	1.6346361637	0.04128289222	0.0288281440	0.0186078548
5	4.9552288055	1.2898740768	1.6175961494	0.0451571941	0.0331487655	0.0188508033
6	5.0490729808	1.2546651363	1.6251549720	0.0425729751	0.0259521007	0.0162870883
7	5.0559189319	1.2491188049	1.6198101043	0.0402898788	0.0273351669	0.0176029205
8	5.0879919528	1.2580881118	1.6260371208	0.0426468849	0.0263381004	0.0170559883
9	5.0328917503	1.2491509914	1.6144649982	0.0430219173	0.0329370498	0.0176239013
10	5.1429288387	1.2202110290	1.5727391242	0.0396611690	0.0257260799	0.0160610675
Ср. зн.	5.0848807811	1.2474863290	1.6098655700	0.0418768405	0.0280930280	0.0171558380

Домашнее задание

Задача №3750. Количество совпадающих

Даны два списка чисел, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.

Примечание. Эту задачу на Питоне можно решить в одну строчку.

Входные данные

Вводятся два списка чисел. Все числа каждого списка находятся на отдельной строке.

Выходные данные

Выведите ответ на задачу.

Примеры

входные данные
1 3 2 4 3 2
выходные данные
2

Задача №365. Заполнение спиралью

Дано число `n`. Создайте массив `A[2*n+1][2*n+1]` и заполните его по спирали, начиная с числа `0` в центральной клетке `A[n+1][n+1]`. Спираль выходит вверх, далее закручивается против часовой стрелки.

Входные данные

Программа получает на вход одно число `n`.

Выходные данные

Программа должна вывести полученный массив, отводя на вывод каждого числа ровно 3 символа.

Примеры

входные данные
2
выходные данные
12 11 10 9 24 13 2 1 8 23 14 3 0 7 22 15 4 5 6 21 16 17 18 19 20

Задача №1099. Скидки

В супермаркете проводится беспрецедентная акция – «Покупая два любых товара, третий получаешь бесплатно*», а внизу мелким шрифтом приписано «* - из трех выбранных вами товаров оплачиваются два наиболее дорогих».

Вася, идя в супермаркет, определился, какие товары он хочет купить, и узнал, сколько они стоят. Помогите ему определить минимальную сумму денег, которую ему нужно взять с собой, чтобы в итоге стать счастливым обладателем этих товаров.

Входные данные

Во входном файле задано сначала число N ($1 \le N \le 1000$), а затем N чисел – стоимости выбранных Васей товаров. Все стоимости – натуральные числа, не превышающие 10000.

Выходные данные

В выходной файл выведите одно число – сумму денег, которую Вася должен взять с собой в супермаркет (минимально возможную).

Комментарии к примерам тестов

1. Вася сначала пройдет через кассу с товарами стоимостью 1, 3 и 4 – заплатит 7 рублей и товар стоимостью 1 получит в подарок, а затем снова зайдет в супермаркет и купит товары стоимостью 5 и 7, еще один товар стоимостью 5 получив в подарок.

2. Вася в первый заход в супермаркет купит товары стоимостью 15 и 25 рублей, в качестве подарка взяв товар стоимостью 8 рублей. А во второй заход в супермаркет купит товары стоимостью 3 и 8, не взяв никакого подарка.

Примеры

входные данные
6 1 5 4 3 5 7
выходные данные
19

входные данные
5 3 15 25 8 8
выходные данные
51

АПЛАНА
ГРУППА КОМПАНИЙ

Входит в ГК Аплана

АКАДЕМИЯ АЙТИ

Основана в 1995 г.

E-learning и очное обучение

Направления обучения:

- Информационные технологии
- Информационная безопасность
- ИТ-менеджмент и управление проектами
- Разработка и тестирование ПО
- Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск, Хабаровск, Красноярск, Тюмень, Нижний Новгород, Краснодар, Волгоград, Ростов-на-Дону

Головной офис в Москве

Разработка программного обеспечения и информационных систем

Ежегодные награды Microsoft, Huawei, Cisco и другие

Программы по импортозамещению

Ресурсы более 400 высококлассных экспертов и преподавателей

Сеть региональных учебных центров по всей России

Крупные заказчики

СБЕР

РЖД

Билайн

TATNEFT

НОРНИКЕЛЬ

100+
сотрудников

АКАДЕМИЯ АЙТИ

Спасибо за внимание!

Центральный офис:
Москва, Варшавское шоссе 47, корп. 4, 7 этаж
Тел: +7 (495) 150-96-00
academy@it.ru
academyit.ru