

1. Задачи на обработку последовательности

В следующих задачах предполагается, что в файле записана последовательность чисел неизвестной длины (возможно, пустая). Требуется за один просмотр файла и без запоминания последовательности в массиве определить требуемую характеристику последовательности.

Программа должна содержать функцию, которая получает в качестве параметра имя файла (или указатель на файл) и возвращает требуемое значение.

При автоматизированном тесте программы обычно предполагается, что файл с данными и файл для результата имеют заранее заданные имена (например, input.txt, output.txt).

Функция main открывает необходимые файлы, проверяет успешность открытия, обращается к функции для вычисления результата и выводит результат в соответствующий файл.

Как вариант задания, имена файлов могут вводиться с клавиатуры, а результат печататься на экран.

1. Подсчитать среднее арифметическое чисел из последовательности.
2. Подсчитать среднее геометрическое чисел из последовательности.
3. Подсчитать среднее гармоническое чисел из последовательности.
4. Подсчитать количество чисел, больших предыдущего.
5. Определить есть ли в последовательности число X (для вещественных чисел — с точностью ε).
6. Определить номер последнего числа, равного X (для вещественных чисел — с точностью ε).
7. Определить все ли элементы последовательности равны между собой (для вещественных чисел — с точностью ε).
8. Определить является ли последовательность возрастающей, убывающей?
9. Определить удовлетворяют ли элементы последовательности данному рекуррентному соотношению $c_1 a_{i-1} + c_2 a_i + c_3 a_{i+1} = b$ (для вещественных чисел — с точностью ε).
10. Определить количество различных элементов целой неубывающей последовательности.
11. Определить общее количество элементов в постоянных участках целой последовательности.
12. Определить порядковый номер первого числа, равного максимуму по всей целой последовательности.
13. Определить номер последнего числа, равного минимуму по всей целой последовательности.
14. Определить количество чисел, равных минимальному из всей целой последовательности.
15. Найти среднее квадратическое отклонение от среднего арифметического. $D = \frac{1}{n} \sum_{i=1}^n (x_i - M)^2$, где M — среднее арифметическое.
16. Найти величину максимального отклонения элементов последовательности от их среднего арифметического.
17. Найти количество возрастающих участков последовательности.
18. Найти сумму четных элементов во всех возрастающих участках целой последовательности.
19. Определить каких участков в последовательности больше — возрастающих или невозрастающих.
20. Найти количество элементов в наибольшем постоянном участке целой последовательности.
21. Найти количество элементов в постоянном участке целой последовательности с наибольшей суммой элементов этого участка.
22. Найти длину возрастающего участка последовательности с наибольшим количеством элементов.
23. Найти наибольшую сумму возрастающего участка последовательности (т.е. максимум из сумм элементов по каждому возрастающему участку).
24. Найти среднее арифметическое локальных экстремумов последовательности.
25. Найти максимальное расстояние (количество элементов) между двумя соседними локальными минимумами последовательности.
26. Найти среднее арифметическое значений элементов целой последовательности, учитывая значения в постоянных участках только один раз.
27. Найти среднее арифметическое, взяв по одному элементу из каждого постоянного участка целой последовательности.
28. Найти максимальную сумму подряд идущих элементов последовательности.
29. Последовательность чисел представляет собой коэффициенты многочлена по возрастанию степеней. Вычислить многочлен и его производную в заданной точке x .
30. Последовательность чисел представляет собой коэффициенты многочлена по убыванию степеней. Вычислить многочлен и его производную в заданной точке x .

Замечание 1. Для простоты можно считать, что локальный экстремум (максимум, минимум) — это элемент последовательности строго больше (меньше) своих соседей. Таким образом, первый или последний элемент последовательности не могут быть локальными экстремумами.

Замечание 2. Для зачетов и контрольных работ формулировки задач могут быть немного модифицированы, не нарушая базовых идей алгоритма решения. Например, минимум может быть заменен на максимум, возрастание на убывание и т.п.

Задачи для 1 курса (2011-2012 уч. год.)

2. Задачи на работу с массивами

Решения следующих задач должны содержать функцию, которая получает в качестве параметров имя массива и его длину (или нескольких массивов, если этого требуют условия задачи) и выполняет необходимые действия.

При решении не разрешается создавать или резервировать в программе дополнительную память, соизмеримую по размерам с объемом исходных данных. То есть, нельзя создавать дополнительные массивы, если это явно не оговорено в задаче.

Функция `main` должна заполнить массив числами из файла. Для определения длины массива предусматривается два варианта: 1) по значению первого числа в файле, 2) непосредственным подсчетом количества чисел в файле. Результат также выводится в файл.

0. Все задачи из предыдущего списка (для последовательностей) могут быть переформулированы для массивов.
1. Симметричны ли значения элементов массива целых чисел?
2. Переставить элементы массива в обратном порядке.
3. Циклически сдвинуть элементы массива на одну позицию вправо.
4. Сравнить два неупорядоченных целочисленных массива A и B как числовые множества без повторения элементов: $A = B$ и $A \subset B$.
5. Для двух целочисленных массивов построить третий массив, являющийся их объединением как числовых множеств без повторения элементов. Указать длину получившегося массива.
6. Для двух целочисленных массивов построить третий массив, являющийся их пересечением как числовых множеств без повторения элементов. Указать длину получившегося массива.
7. Определить какое число встречается в массиве целых чисел наибольшее количество раз.
8. Удалить из целочисленного массива одинаковые значения, т.е. если какое-то значение встречается несколько раз (в разных местах массива), то оставить только первый такой элемент, а остальные удалить из массива. Оставшиеся элементы сдвинуть к началу массива, и указать их количество.
9. Сократить подряд идущие одинаковые элементы целочисленного массива до одного элемента. То есть, если в массиве встречается несколько одинаковых элементов, стоящих рядом, то оставить только один из них, а остальные удалить из массива. Оставшиеся элементы сдвинуть к началу массива, и указать их количество.
10. Удалить из массива все отрицательные значения, а оставшиеся уплотнить (сдвинуть) с сохранением исходного порядка к началу массива. Указать количество оставшихся значений.
11. Циклически сдвинуть элементы массива на K позиций вправо с затратой $O(N)$ действий (N -длина массива)
12. Каждый элемент $a[i]$ массива заменить на сумму элементов исходного массива вплоть до него самого включительно, т.е. от 0 до i -го.
13. Каждый элемент массива заменить на полусумму его соседних элементов (кроме первого и последнего)
14. Назовем x -отрезком группу подряд идущих элементов массива, каждый из которых равен x . Для заданного числа x заменить элементы каждого x -отрезка на полусумму элементов, прилегающих к этому отрезку справа и слева. Если x -отрезок расположен в начале или конце массива, считать второй крайний элемент равным нулю.
15. Сгруппировать положительные элементы массива в его начале, а отрицательные — в конце с сохранением их порядка.
16. Назовем массив из N целых чисел счастливым, если существует такое $0 < k < N$, что сумма элементов с индексами от 0 до $k-1$ совпадает с суммой элементов с индексами от k до $N-1$. Определить является ли данный массив счастливым.
17. Назовем массив из целых чисел плотным, если множество значений элементов массива полностью заполняет некоторый отрезок $[a, b]$ (рассматриваются целые значения). Определить является ли данный массив плотным.
18. Получить массив биномиальных коэффициентов для степени N , последовательно вычисляя строки треугольника Паскаля (можно использовать только один массив).
19. Элементы массива не убывают. Двоичным поиском определить позицию, где в этот массив можно вставить данное число x .
20. Даны два неубывающих массива. Построить третий неубывающий массив, который является объединением первых двух (элементы могут повторяться).
21. Выполнить следующее преобразование. Элементы с четными индексами сгруппировать в начале массива с сохранением их исходного порядка относительно друг друга, а элементы с нечетными индексами сгруппировать в конце массива также с сохранением их исходного порядка.
22. Выполнить следующее преобразование массива длины N . Элементы с индексами $i \leq [(N+1)/2]$ переместить на позиции с четными индексами с сохранением их исходного порядка относительно друг друга, а оставшиеся элементы ($i > [(N+1)/2]$) разместить на позициях с нечетными индексами также с сохранением их исходного порядка. Т.е. начальная и конечная половины массива “перемешиваются” чередованием элементов.
23. Пусть в массиве последовательно записаны цифры некоторого длинного десятичного числа. Реализовать функции “прибавляющие единицу” и “вычитающие единицу” из такого числа. (для реализации переноса из “старшего разряда” можно заранее запасти 1 лишний элемент в массиве)

Замечание. В задачах, где массив преобразовывается с удалением элементов, предполагается, что ответ (преобразованный массив) должен быть размещен в начале исходного массива. При нужно вычислить количество элементов в новом полученном массиве. Если не сказано специально, то элементы исходного массива, которые остаются в конце исходного массива и не входят в ответ, можно “не чистить”, т.е. их значения могут оставаться любыми.

Замечание. В задачах, где в условии упоминается несколько массивов, предполагается, что в решении используются только эти массивы (дополнительные массивы создавать нельзя) и при этом они имеют длину, необходимую только для размещения исходных данных.

Замечание. На данном этапе при решении этих задач вопрос о вычислительной сложности не имеет очень принципиального значения. То есть, алгоритм с квадратичной оценкой трудоемкости считается вполне допустимым. Вопрос о том можно ли построить алгоритм с лучшей оценкой можно предложить рассмотреть сильным студентам в качестве самостоятельного упражнения.

Задачи для 1 курса (2011-2012 уч. год.)

3. Задачи на сортировку массивов

При решении следующих задач нужно реализовать отдельную функцию для сортировки массива чисел, функцию для проверки массива на упорядоченность, функцию для чтения массива из файла и функцию для генерирования случайного массива указанной длины.

Нужно составить тест для проверки трудоемкости алгоритмов сортировки. Для этого релизуется один из методов “медленной” сортировки, один из методов “быстрой” сортировки и сортировка с использованием библиотечной функции `qsort`. Для заданного N генерируется неупорядоченный массив из N чисел, который сортируется всеми тремя методами. Замеряется время работы каждой сортировки (например, с помощью функции `clock()` или любых других подходящих). Тест повторяется несколько раз с удвоением длины массива N .

Алгоритмы медленной сортировки:

1. Простая сортировка обменов.
2. Пузырьковая сортировка.
3. Сортировка просеиванием.
4. Вставка с последовательным поиском.

Алгоритмы “быстрой” сортировки:

5. Сортировка слиянием.
6. Быстрая сортировка (quicksort).
7. Пирамидальная сортировка (heapsort).
8. Линейная сортировка (подсчетом) целого массива.
9. Сортировка целого массива группировкой с последовательным упорядочиванием битов.

В следующих задачах требуется реализовать сортировку с использованием “нетрадиционных” функций сравнения элементов массива.

10. Сгруппировать положительные элементы в начале массива, а отрицательные в конце так, чтобы и положительные, и отрицательные элементы внутри своей группы были упорядочены по возрастанию.
11. В данном массиве целых чисел упорядочить по возрастанию элементы с четными значениями, а элементы с нечетными значениями оставить на их исходных местах.
12. Упорядочить элементы массива целых чисел по возрастанию количества единиц в их битовом представлении.

4. Задачи на битовые операции и делимость

1. Написать функции, записывающие 0 или 1 в указанный бит данного целого числа и оставляющие остальные биты без изменения.
2. Проверить четность количества единиц в двоичном представлении данного целого числа.
3. Найти первые N целых чисел, у которых младший байт является зеркальным отражением следующего байта.
4. Определить позицию самой старшей единицы в битовом представлении данного целого числа.
5. Написать функции, позволяющие работать с целым числом (long) как с “массивом” четырех байтовых чисел (char).
6. Написать функцию, которая зеркально переворачивает битовое представление целого числа (аналогично перестановке массива в обратном порядке).
7. Написать функцию, которая циклически сдвигает битовое представление целого числа на указанное количество позиций вправо или влево (аналогично циклическому сдвигу массива).
8. Написать функцию, которая переставляет байты целого числа (int) в обратном порядке.
9. Возвести число в степень N за не более чем $2 \log_2 N$ умножений.
10. Вывести в файл все подмножества множества $\{1, \dots, N\}$.
11. Вывести в файл все k -элементные подмножества мн-ва $\{1, \dots, N\}$.
12. Найти наибольший общий делитель двух целых чисел (алгоритм Евклида).
13. Определить четность произвольной перестановки N чисел.
14. Вычислить первые N простых чисел.
15. Разложить натуральное число на простые множители.
16. Вывести значение целого числа N в “словесной форме”.
17. Вычислить представление числа $1/N$ в виде десятичной дроби (начало и период) (то же для числа M/N).

5. Задачи на обработку множества точек

В следующих задачах предполагается, что в файле записано несколько пар чисел, которые можно рассматривать как координаты множества точек на плоскости или как координаты множества концов отрезков на прямой.

1. Множество точек определяет ломаную. Имеет ли она самопересечения?
2. Множество точек определяет многоугольник. Является ли он выпуклым?
3. Множество точек определяет многоугольник. Определить угол, под которым данный многоугольник виден из заданной точки.
4. Множество точек определяет многоугольник. Для данной точки определить где она расположена относительно этого многоугольника: внутри, снаружи, на границе.
5. Множество точек определяет выпуклый многоугольник. Найти его минимальный и максимальный диаметры (вписанная и описанная окружности).
6. Дано множество отрезков на прямой. Покрывает ли объединение этих отрезков заданный отрезок $[a, b]$?
7. Два множества точек задают два многоугольника (без самопересечений, но не обязательно выпуклые). Определить расстояние между этими многоугольниками.
8. Два множества точек задают два выпуклых многоугольника, не лежащих один внутри другого. Определить расстояние между этими многоугольниками с линейной оценкой трудоемкости по суммарному количеству вершин.
9. Дано множество точек. Найти центр и радиус минимального круга, который содержит все эти точки.
10. Дано множество отрезков на прямой. Выбрать из него и вывести те отрезки, объединение которых дает отрезок наибольшей длины.
11. Даны центры равномерно растущих кругов на плоскости. При столкновении друг с другом столкнувшиеся круги прекращают свой рост. Найти радиусы кругов, когда процесс роста остановится полностью.
12. Дано множество точек на плоскости. Построить выпуклую оболочку этого множества.
13. Множество точек определяет выпуклый многоугольник. Построить многоугольник, который получится, если линию, задающую каждую сторону, отодвинуть в перпендикулярном ей направлении на величину h .