

Задачи для 1 курса (2020-2021 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 заполнения тестового массива указанной длины с заданными свойствами (случайный, упорядоченный, частично упорядоченный, “волнообразный” и т.п.).

Нужно составить тест для проверки трудоемкости алгоритмов сортировки. Для этого реализуется 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 . Сравнение элементов реализуется внешней функцией, которая передается в процедуру сортировки через параметр указатель на функцию.

Дополнительно реализуется и тестируется какой-либо метод специальной сортировки.

Алгоритмы медленной сортировки:

1. Простая сортировка обменов (перестановка максимума).
2. Пузырьковая восходящая сортировка.
3. Сортировка просеиванием (нисходящая пузырьковая).
4. Сортировка просеиванием с бинарным поиском позиции вставки.

Алгоритмы “быстрой” сортировки:

5. Сортировка слиянием с доп. памятью $O(N)$.
 6. Сортировка слиянием без доп. памяти с трудоемкостью $O(N \log^2 N)$.
 7. Быстрая сортировка (quicksort).
 8. Пирамидальная сортировка (heapsort).
- “Специальные” сортировки:*
9. Сортировка подсчетом для целого массива.
 10. Сортировка массива целых чисел битовым аналогом быстрой сортировки.
 11. Сортировка массива целых чисел битовым radix методом с сортировкой подсчетом.
 12. Сортировка массива целых чисел битовым radix методом.

4. Задачи на битовые операции и делимость

1. Определить позицию самой старшей единицы в битовом представлении данного целого числа.
2. Написать функции, записывающие 0 или 1 в указанный бит данного целого числа и оставляющие остальные биты без изменения, а также проверяющие значение отдельного бита числа (битовое множество).
3. Проверить четность количества единиц в двоичном представлении данного целого числа.
4. Найти первые N целых чисел, у которых младший байт является зеркальным отражением следующего байта.
5. Написать функции, позволяющие работать с целым числом (long) как с “массивом” четырех однобайтовых чисел (char).
6. Написать функцию, которая зеркально переворачивает битовое представление целого числа (аналогично перестановке массива в обратном порядке) в пределах до старшей значащей единицы.
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. В заданном диапазоне найти все числа с симметричной битовой записью в пределах до старшей значащей единицы.
10. Возвести число в степень N за не более чем $2 \log_2 N$ умножений.
11. Вывести в файл все подмножества множества $\{1, \dots, N\}$.
12. Вывести в файл все k -элементные подмножества мн-ва $\{1, \dots, N\}$.
13. Найти наибольший общий делитель двух целых чисел (алгоритм Евклида).
14. Вычислить первые N простых чисел.
15. Разложить натуральное число на простые множители.
16. 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 (габариты).
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. Дан прямоугольник на плоскости и семейство отрезков. Построить отрезки, которые получаются в результате пересечения данных отрезков с прямоугольником (клипирование).
10. Дано множество точек. Найти центр и радиус минимального круга, который содержит все эти точки.
11. Дано множество отрезков на прямой. Выбрать из него и вывести те отрезки, объединение которых дает отрезок наибольшей длины.
12. 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.
13. Дано множество точек на плоскости. Построить выпуклую оболочку этого множества.
14. Множество точек определяет выпуклый многоугольник. Построить многоугольник, который получится, если линию, задающую каждую сторону, отодвинуть в перпендикулярном ей направлении на величину h .