

~~ЗАДАНИЕ 1~~

(срок сдачи 21–26 октября)

В каждой задаче ответьте на вопрос о том, как растет время работы программы и используемая программой память с ростом размера входных данных. Программа должна быть написана на языке С, иметь комментарии, осмысленные имена переменных и функций, быть удобной для чтения, компилироваться без предупреждений, valgrind и address sanitizer не должны находить в ней ошибок.

1. «Max». Написать программу, которая выводит максимальное число из n заданных чисел. В первой строке входа дано число n , а в следующей строке указано n целых чисел.

2. «Числа Фибоначчи». Написать программу, которая по данному n находит n -е число Фибоначчи F_n .

Оценить число элементарных операций, которое необходимо сделать в рекурсивном и нерекурсивном алгоритмах вычисления числа Фибоначчи.

3. «Биномиальные коэффициенты». Написать программу, которая для данных натуральных чисел n и k находит количество сочетаний по k неупорядоченных объектов из n . Оценить, как растет время работы вашей программы с ростом n и k . Постройте вычисления таким образом, чтобы промежуточные значения не превышали ответ.

4. «Простые числа». Написать программу, которая определяет, является ли введенное число простым.

5. «Скобки». Написать программу, которая определяет, является ли введенная скобочная структура правильной. Скобки могут быть открывающие и закрывающие, 4 типов: ([{ < > }])

6. Даны два вектора. Найдите полярный угол между ними.

Входные данные содержат четыре целых числа, по модулю не превосходящих 10^4 координаты двух ненулевых векторов. Выведите величину неориентированного угла между векторами из интервала $[0, \pi]$ с абсолютной погрешностью не хуже 10^{-5} .

7. Пересечение множеств

Во входном файле два множества натуральных чисел.

В выходной файл выведите их пересечение (перечисление элементов через пробел в любом порядке без повторений) или слово empty, если пересечение пусто.

Множества на входе представлены как последовательности натуральных чисел, разделенных пробелом, и завершающиеся числом -1 — идентификатором конца. Возможны повторения элементов, которые надо исключить. Размеры множеств меньше 1000. Сами числа меньше 10^6 .

8. Случайный спуск по дереву

На вход подается описание бинарного дерева. На листьях (висячих вершинах) этого дерева написаны целые числа (от -10^9 до 10^9). Идем от

корня дерева, случайно поворачивая направо или налево (с вероятностями 0.5).

Чему равно среднее значение числа на листе, в который мы в конечном счете придем? Ответ вывести с точностью до двух знаков после запятой. Формат описания дерева следующий:

tree ::= leaf | (tree tree);

leaf ::= integer;

✓ ~~9~~ Дано N целых чисел. Требуется выбрать из них три таких числа, произведение которых максимально.

Дополнения к заданиям для конкретных групп выдаются в виде контестов на <http://judge.mipt.ru>

~~ЗАДАНИЕ 2~~ (срок сдачи 2–7 декабря)

Задачи должны быть написаны на языке С. Должен быть выбран оптимальный (с минимальной вычислительной сложностью) алгоритм. Проверяйте корректность входных данных. Программа должна быть удобной для чтения человеком, иметь единый стиль отступов. Переменные и функции должны иметь осмысленное и понятное название. Программа должна компилироваться без предупреждений и ошибок с опциями -Wall -Wextra -Werror. Address Sanitizer и Valgrind не должны находить ошибок при работе программ. При выделении динамической памяти она должна освобождаться. К программе должны быть предложены тесты, покрывающие различные случаи. Рекомендуется использовать makefile. Части кода, которые можно использовать повторно, рекомендуется выделять в отдельные функции.

✓ ~~Q~~ В строках текстового файла содержатся разделенные пробелом целочисленные координаты вершин прямоугольной односвязной плоской фигуры. Вершины фигуры перечисляются в произвольном порядке. Вычислить площадь заданной фигуры.

+ ~~2.~~ На поле C3 шахматной доски неподвижно располагается белый король. Произвольно на доску выставляются черный король и белый ферзь. Напишите программу, доказывающую, что при любой (имеющей смысл) расстановке двух последних фигур мат черному королю может быть объявлен не более чем за 25 ходов.

✓ ~~3.~~ «Атлеты». Написать программу, которая находит «башню» из атлетов максимальной высоты. Атлеты характеризуются двумя параметрами – массой и силой. Сила равна максимальной массе, которую атлет может держать на плечах. Известно, что если атлет тяжелее, то он точно сильнее.

Входом является число атлетов n и n пар (масса, сила). При написании программы используйте сортировку структур.

4. «Отрезки». Написать программу, которая для множества заданных отрезков находит минимальное подмножество отрезков, объединение которых покрывает отрезок $S = [0, 10000]$. Число отрезков и координаты их концов заданы на входе. Все координаты целочисленные. Подсказка: покрывайте отрезок S пошагово, двигаясь слева направо. На каждом шаге будет непокрытая часть $[x, 10\ 000]$. Из оставшихся отрезков выбирайте тот, который урежет непокрытую часть до $[y, 10\ 000]$, где y максимальное. Решите эту задачу за время $O(n \log n)$, где n – количество данных отрезков.

5. «Записная книжка I». Написать программу, которая реализует функциональность телефонной записной книжки. А именно, из стандартного входа программа получает последовательность команд на добавление (*INSERT*) или поиск (*FIND*) записей. Примеры команд: *INSERT Sidorov 1234567*, *INSERT Ivanov 7654321*, *FIND Sidorov*. При выполнении команды *INSERT* программа добавляет пару (фамилия, номер) в своё хранилище и выводит строку «*OK*», если в хранилище нет записи с такой фамилией, или изменяет соответствующую указанной фамилии запись и выводит строку «*Changed. Old value = X*», если запись с такой фамилией уже есть в хранилище и соответствующий телефонный номер был X . При выполнении команды *FIND* программа выводит телефонный номер для указанной фамилии или выводит «*NO*», если указанной фамилии нет в справочнике. Следует использовать структуры, состоящие из двух элементов *name* и *number*. Использовать технику динамического выделения памяти для хранения записей. Оценить, как в среднем растёт число элементарных операций при выполнении команд *INSERT* и *FIND* с ростом числа хранимых записей. Записи хранить в двоичном дереве поиска.

6. «Записная книжка II». Решите задачу 5, используя хэш-таблицу с разрешением коллизий методом цепочек либо методом открытой адресации.

7. Даны две последовательности латинских букв длиной не более 1024 символов. Найти их наибольшую общую подпоследовательность.

8. Неориентированный взвешенный связный пока что граф с петлями задан списком рёбер, для каждого ребра указана его длина. Веса всех ребер неотрицательны. Найдите минимальное покрывающее дерево.

9. Реализуйте бинарное дерево поиска для целых чисел. Программа получает на вход последовательность целых чисел и строит из них дерево. Элементы в деревья добавляются в соответствии с результатом поиска их места. Если элемент уже существует в дереве, добавлять его не надо. Балансировка дерева не производится.

На вход программа получает последовательность натуральных чисел меньших 10000. Последовательность завершается числом 0, которое означает конец ввода, и добавлять его в дерево не надо.

Выведите единственное число – высоту получившегося дерева.

10. «Задача Иосифа». Пусть n человек, стоящие по кругу, считаются (начиная с первого) считалкой из m слов. Человек, на котором считалка заканчивается, – выбывает, круг смыкается, а счет продолжается с человека, следующего за выбывшим. Напишите программу, выводящую номера трех человек, выбывших последними, в порядке их выбывания. При написании программы следует реализовать список с использованием динамического выделения и освобождения памяти.

Дополнения к заданиям для конкретных групп выдаются в виде контестов на <http://judge.mipt.ru>