Командная олимпиада “Высшая проба” 2019

Разбор задач

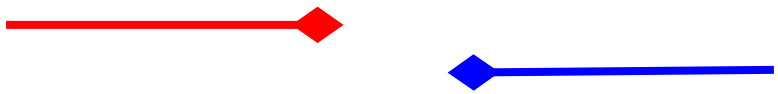
Задача A. Покраска деревьев

Тема: Пересечение отрезков

Нужно понять, пересекаются ли отрезки или нет. Проще всего упорядочить отрезки (например, по заданному нам центру) и проверить, что правая граница левого отрезка больше либо равна левой границе правого. Тогда есть пересечение и ответ: большая из правых границ минус меньшая из левых границ отрезков.

На рисунке красный - это “левый” отрезок, а синий - правый

В противном случае отрезки не пересекаются и нужно просто сложить их длины.



Задача B. Продавец рыбы

Тема: Линейный поиск

Переберём каждый из N дней в качестве дня, в который мы покупаем рыбу. Определим максимальную прибыль, которую можно получить при покупке в этот день. Для этого среди следующих K чисел найдём максимальное и посчитаем разность. Среди всех таких разностей найдём максимальную - она и будет ответом.

Сложность решения: O(NK) ~ 1 000 000 000 операций

Задача C. Сумма номеров

Тема: Два указателя

Переберём все левые границы отрезка. Для каждой левой границы можно перебирать все возможные правые, но это долго $O(N^2)$ (а можно придумать и за $O(N^3)$).

Зафиксируем левую границу в начале последовательности и будем двигать правую до тех пор, пока сумма не станет больше или равна K. Если равна - прибавим к ответу 1. Нулей в последовательности нет, поэтому с такой левой границей это единственный отрезок. Передвигаем левую границу на 1, вычитаем вышедшее за пределы отрезка число, а правую границу двигаем вправо от текущего положения (и прибавляем попавшие в отрезок числа), до тех пор, пока сумма меньше K. Правая граница никогда не может быть сдвинута вправо после сдвига левой границы на 1.

Задача D: Контейнеры

Темы: Эвристические методы, моделирование

У этой задачи существует конструктивное решение:

1. Берём два контейнера AB и перемещаем их на свободное место
2. Берём два последних контейнера BB (или два первых контейнера AA) и ставим их на свободное место

Повторяем шаги до тех пор, пока контейнеры не кончатся. На последнем шаге пункт 2 не выполняем.

Можно брать контейнеры не из конца (или начала) последовательности, но, обычно, это сложнее.

Количество операций: 2N - 3 за исключением ситуации N = 1: в этом случае ответ равен 0.

Задача E: Пираты Баренцева моря

Темы: Сортировка, жадный алгоритм, простой перебор

Отсортируем корабли по Y координате. Переберем все столбцы, в которых будем выстраивать корабли. Идём по всем возможным координатам Y, и расставляем корабли на эти клетки так, как они идут в отсортированной последовательности.

Для определения количества ходов до корабля достаточно просуммировать модули разности координат для исходной и целевой клетки. О проходе корабля сквозь другой и порядке их X координат можно не беспокоиться, т.к. Проход корабля сквозь другой эквивалентен по количеству заменой старого корабля на новый.

Задача F: Рыбалка

Темы: Быстрая сортировка, моделирование

Определим, какая из лодок “нижняя”, а какая “верхняя” и введём три типа “событий” (координат клеток): клетка с рыбой, изменение Y координаты нижней лодки и изменение Y координаты верхней лодки.

Отсортируем все события по X, а при равных X - по Y. Найдём событие “начало движения лодок” и начиная с этого момента будем идти поддерживать актуальные Y координаты нижней и верхней лодок (они изменяются при возникновении соответствующих событий). Если событие - клетка с рыбой, то проверяем, попадает ли её Y в промежуток между нижней и верхней лодкой, и если да - прибавляем к ответу 1.

Вместо сортировок можно было использовать и бинарное дерево поиска.

Задача G: Квадропалиндром

Темы: Перебор, комбинаторика, работа со строками

Задача является задачей построения паросочетания в произвольном графе, однако из-за небольших ограничений может быть решена полным перебором. Сгенерируем все перестановки строк и для каждой проверим, является ли она квадропалиндромом.

Жадный алгоритм, когда мы берём первые попавшиеся совместимые строки и навсегда объединяем их в пару, может не работать, например, в случае если будут сопоставлены строки 1 и 2 из такого примера:

1234

????

1234

5678

Задача H: Стеганография

Темы: Обработка текста, двочиные числа

Для решения задачи удобно перевести весь текст в нижний (или верхний) регистр.

После этого достаточно идти по всем позициям в тексте подряд и проверять, встречается ли начиная с этой позиции слово one или zero. Также необходимо создать счетчик для ответа, в начале равный нулю. Появление one должно заменять K = K * 2 + 1, а появление zero K = K * 2.

В случае использования C++ или Pascal необходимо использование 64-битных переменных.

Задача I: Строки Фибоначчи

Темы: Динамическое программирование, рекуррентные соотношения

Для каждой позиции в строке легко определить, есть ли начаиная с неё F(1, X, Y), F(2, X, Y) и F(3, X, Y), удобно также отдельно посчитать F(4, X, Y).

Обозначим за G(pos, i, X, Y) функцию, возвращающую истину, если начиная с позиции pos идет F(i, X, Y). Тогда G(pos, i + 1, X, Y) = G(pos, i, X, Y) and G(pos + fib(i), i - 1, X, Y), где fib(i) - i-ое число Фибоначчи (совпадающее с длиной i-ой строки Фибоначчи). Достаточно посчитать эти функции в порядке возрастания i. Можно делать это эффективно, не перебирая X, Y, а определяя их по первым буквам.

Сложность такого решения O(NlogN), т.к. числа Фибоначчи растут примерно как 2^N

Также возможно решение с использованием хешей.

Задача J: Как белка в колесе

Темы: Представление графов, конечные автоматы, динамическое программирование

Составим и будем постепенно заполнять таблицу, где строками будет позиция в тексте, а столбцами - текущей номер команды. Будем моделировать выполнение программы, запоминая при этом все состоянии, в которых мы побывали. Если выполнение программы закончилось или мы пришли в состояние, для которого уже известен результат, то для всех запомненных состояний запишем этот результат.

Критерием бесконечного цикла является попадание в состояние, для которого уже известно, что из него мы попадаем в бесконечный цикл, либо попадание в одно из состояний, в котором мы уже побывали, но еще не знаем результата. Это означает, что мы пришли из него в него же и образовался бесконечный цикл.

Задача K: Барбершоп

Темы: Моделирование, приоритетная очередь

В этой задаче достаточно было создать очередь с приоритетами из парикмахеров, где приоритетом является время последней стрижки (меньше - ближе к началу очереди), а в случае совпадения времени - номер барбера (опять же меньше - ближе к началу очереди). При использовании такого подхода достаточно было просто брать барбера из начала приоритетной очереди, удалять его, пересчитывать время окончания стрижки и снова добавлять в приоритетную очередь.

Решение задачи возможно также и с обычными очередями, но требует аккуратной обработки случаев одновременного окончания стрижки несколькими барберами (накопление всех одновременно заканчивающих и их сортировка).