

Краткий разбор задач Московской олимпиады 10-11 классов, 2017 г.

Задача А. Одномерные мемы

Темы: поиск подстроки в строке, разбор случаев, обработка строк, накопление суммы

В этой задаче требовалось найти вхождение подстроки, закодированной алгоритмом сжатия RLE в также закодированную строку.

Для облегчения решения задачи исходную строку (и подстроку) необходимо было разбить на последовательность пар (число, буква). После этого необходимо рассмотреть три случая:

- 1) Если подстрока состоит только из одной пары, то необходимо было проверить её вхождение обычным линейным поиском. Буква должна совпадать, а число в подстроке быть меньше либо равным, чем число в строке.
- 2) Если подстрока состоит из двух пар, то необходимо было проверять всевозможные пары двух подряд идущих элементов строки, каждый аналогично п.1.
- 3) Если подстрока состоит более чем из двух пар, то необходимо было отбросить крайние пары подстроки и осуществить поиск оставшейся подстроки в строке. Если оставшаяся подстрока была найдена, то необходимо проверить, что отброшенные по краям пары удовлетворяют условию из п.1.

Для восстановления ответа необходимо было накапливать сумму чисел в пройденных при поиске парах строки. В большом тесте эта сумма могла быть достаточно большой, и для её хранения требовался 64-битный тип данных.

Задача В. Снос пятиэтажек

Темы: быстрая сортировка, сортировка структур, жадный алгоритм, эффективный перебор, суммы на префиксах

В этой задаче необходимо было построить не менее заданного количества квартир, используя три типа домов разной вместительности и стоимости, и покупая под них участки.

Если количество участков, необходимых для застройки, известно и равно K , то под застройку необходимо купить K наиболее дешевых участков. Для этого удобнее всего отсортировать участки по убыванию стоимости, сохраняя их номера в исходной последовательности для восстановления ответа.

На первый взгляд задача напоминает задачу о рюкзаке: у каждого дома есть вместительность и стоимость аналогично весу и стоимости предмета в задаче о рюкзаке. Однако вместительности домов могли быть очень большими, что не позволяет использовать алгоритм решения задачи о рюкзаке.

Для решения задачи можно было перебрать два некоторых параметра, например, количество домов 1 и 2 типа. Зная эти числа, можно понять, сколько домов третьего типа нужно построить – для этого необходимо понять, сколько квартир осталось построить и вычислить количество домов

третьего типа, разделив количество оставшихся квартир на вместимость дома с округлением вверх. По известному суммарному количеству домов также можно было посчитать суммарное количество участков, необходимых для застройки, и выбрать из них самые дешевые. Для эффективности решения, такую сумму нужно было не пересчитывать каждый раз заново, а заранее подсчитать для всех возможных количеств участков за линейное время.

Каждый из тестов большого мультитеста содержал не более 10^5 участков, таким образом, сложность решения для всего мультитеста могла достигать до 7×10^{11} , однако эффективные отсечения позволяли значительно сократить эту сложность, а время тура позволяло провести вычисления даже без оптимизаций.

Задача C. Рейтинг школ

Темы: максимальное паросочетание в двудольном графе

В этой задаче необходимо было сопоставить каждому школьнику предмет таким образом, чтобы суммарное количество различных предметов, сопоставленных школьникам одной школы, было максимальным.

Наиболее простое решение этой задачи заключалось в применении алгоритма Куна, описание которого можно прочитать, например, здесь: http://e-maxx.ru/algo/kuhn_matching. При этом для повышения эффективности нужно было правильно распределить вершины по долям графа. Поскольку количество предметов было невелико, а школьников могло быть достаточно много, то большей эффективности можно было достичь при помещении школьников в левую долю графа, а предметов – в правую.

Для вычисления максимального паросочетания также можно было воспользоваться алгоритмом поиска максимального потока в графе. Также для решения задачи можно было применить различные эвристические методы, которые могли набирать различные баллы вплоть до полного.

Задача D. Таблицы истинности

Темы: алгебра логики, сортировка подсчетом, множества

В этой задаче необходимо было подсчитать, на скольких наборах переменных, логическое выражение принимает истинное значение.

Для этого требовалось проанализировать вид выражений и выделить несколько типов выражений:

- 1) Состоящие из конъюнкций некоторых переменных и их отрицаний: ответ равен 2^k в степени количество неиспользованных в выражении переменных
- 2) Состоящие из дизъюнкций некоторых переменных и их отрицаний: ответ равен $2^k - 1$ в степени количество неиспользованных переменных.
- 3) Состоящие из конъюнкций всех переменных, объединенных дизъюнкциями. Ответ равен количеству конъюнкций (количеству знаков дизъюнкции плюс один).
- 4) Состоящие из конъюнкций всех переменных, кроме нескольких, объединенных дизъюнкциями. Ответ равен $2^k - 1$ в степени количество неиспользованных переменных умножить на количество конъюнкций.

Такие же типы выражений встречались в большом тесте. Для определения типа выражения удобно было написать программу, определяющую тип выражения с использованием идей сортировки подсчетом или множеств.

Задача Е. Лишние пробелы

Темы: обработка текста, конечные автоматы, полный перебор с отсечениями, множества

В задаче необходимо было превратить группы подряд идущих пробелов на один пробел за наименьшее количество замен вида «замени k подряд идущих пробелов на 1».

Перед применением алгоритма решения задачи необходимо было создать из входного текста одномерный массив, каждый элемент которого равен числу подряд идущих пробелов. Для этого удобно воспользоваться идеями, применяемыми в конечных автоматах. Автомат для обработки имеет два состояния: находится ли он внутри слова или обрабатывается группа пробелов. Если автомат находится внутри слова и очередной символ – пробел, то он переходит во второе состояние и счетчик пробелов устанавливается в единицу. Если автомат находится во втором состоянии и очередной символ – пробел, то счетчик пробелов увеличивается на единицу. Если автомат находится во втором состоянии и встречает символ, отличный от пробела, то накопленное в счетчике количество пробелов добавляется в массив и автомат переходит в первое состояние.

Несложно заметить, что группы пробелов одинаковой длины будут заменены на один одним и тем же набором операций замены. Поэтому достаточно оставить лишь по одному вхождению каждого числа пробелов. Это можно сделать с помощью структуры данных «множества» или с помощью сортировки массива и одного линейного прохода по нему, сравнивая текущий элемент с предыдущим.

Чтобы определить, какой из вариантов замены будет кратчайшим, необходимо было реализовать полный перебор. Поскольку во входных данных количество подряд идущих пробелов не превосходило 1000, то можно понять, что всего потребуется не более 10 операций замены (все 10 операций могут быть операциями замены двух пробелов на 1). Такие небольшие ограничения говорят о том, что может быть организован полный перебор. При этом эффективно сначала пробовать замены маленьких чисел, т.к. они повлияют на наибольшее количество групп пробелов различной длины. В случае если текущее количество операций превышает наилучший на текущий момент ответ, то дальнейший перебор для этой ветви можно было прекратить.

Такой перебор работал достаточно быстро и не нуждался в дополнительных оптимизациях.

Задача Е. Коробочки

Темы: динамическое программирование: один параметр

Эту задачу было удобно решать методом динамического программирования. А именно, для каждого квадрата картона необходимо было посчитать, коробочки какого максимального объема можно сделать из него. Для этого необходимо было для каждого возможного размера квадрата картона перебрать все возможные значения размеров четырех квадратов, отрезаемых по углам, вычислить объём получившейся коробки (её ширина и длина равна размеру большого квадрата минус удвоенный размер отрезаемого квадрата, а высота равна размеру отрезаемого квадрата), и

выбрать среди всех вариантов наилучший. При этом надо учитывать, что отрезаемых по углам квадратов четыре.

После того, как все размеры однажды подсчитаны, можно было запомнить их и пользоваться запомненным результатом для всех тестов из мультитеста. Для каждого прямоугольника необходимо было перебрать все возможные значения отрезаемых квадратов (от 1 до половины длины меньшей из сторон). Размер коробки, получающейся из прямоугольника с вырезанными углами, считается аналогично случаю для квадратных листов из предыдущего абзаца. Также аналогично считаются объёмы коробочек, получающихся из отрезанных квадратов.

Сложность такого решения составляет $O(N^2)$. Размеры листов картона были ограничены 10^5 , поэтому вычисление ответа занимало до нескольких минут.

Различные жадные алгоритмы или математические идеи давали неточный ответ из-за дискретности задачи, однако получали до половины баллов.