Рекомендации к решению заданий

Задача 1. Гирлянда

Так как Алиса может вывесить гирлянду с нужной суммой, число N кратно D. То есть $N = D\cdot (111x+11y+z)$. Вычислить x, y, z можно, находя по порядку целые части и остатки от деления на 111 и 11. Ответ равен $x+ y+z-1$.

Задача 2. Реклама

Для решения задачи нужно аккуратно посчитать максимумы и общие суммы голосов в актерском рейтинге. Затем проверить критерий доверия для каждого актера и рекламируемого им автомобиля. Для этого подсчитать количество стран, в которых набранные ими голоса равны максимумам в стране, и проверить, что процент таких стран не меньше 30%.

Задача 3. Большой теннис

Посчитаем количество последовательностей с выигрышами игроков, приводящими к заданному исходу для каждого сета в отдельности. Пусть в текущем сете $P>V$. Всего сыграно $P+V$ геймов, последний гейм выигрывает первый игрок. Посчитаем, сколько есть вариантов выбора номеров геймов, которые выиграл второй игрок:

- при V=0 только один вариант, никаких выигрышей второго игрока;
- при V=1 число различных вариантов равно P;
- при V=2 выбрать пару номеров геймов, выигрышных для второго игрока, можно $(P+V-1)\cdot(P+V-2)$ способами. Так как пара с одинаковыми номерами посчитается дважды, то ответ равен $(P+V-1)\cdot(P+V-2)/2$.
- при $V>2$. Если P не равно 7, тогда количество различных последовательностей с выигрышами игроков для гейма равно C_{P+V-1}^{V}. При P=7 нетрудно понять, что количество различных последовательностей с выигрышами игроков для V=5 равно C_{5+5}^{5}, а для V=6 равно $2\cdot \mathrm{C}_{5+5}^{5}$.

Ответ на задачу равен произведению количеств последовательностей выигрышей геймов для трех сетов.

Задача 4. Подготовка к олимпиаде

Во входных данных для каждой из N тем X дан список тем, нужных для освоения темы X. Удобно представить эти данные в массивах:

1. массив из N элементов *number_of_themes* для хранения количества связанных тем;
2. массив *list_of_nodes*, *i*-й элемент ссылается на список тем, для которых тема *i* является связанной.

Рассмотрим по очереди темы, для которых все связанные темы уже пройдены. Начальная очередь — список базовых тем, сложность которых равна нулю. Для каждого рассматриваемого элемента очереди *v* уменьшаем на единицу количество связанных тем для элементов из *list_of_node[v]*. Если при этом количество связанных тем для элемента стало равно нулю, то помещаем его в конец очереди и сложность устанавливаем на единицу больше сложности *v*.

Рассмотрение очереди заканчивается, когда у первого элемента очереди сложность будет равна требуемой. Легко понять, что в этот момент у всех элементов очереди сложность равна требуемой. Элементы очереди сортируются по возрастанию и выводятся.

Для хранения информации о связанных темах вместо массива *list_of_nodes* можно использовать двухмерный массив $N \times N$. Вместо использования очереди можно в цикле в массиве со сложностью тем находить элементы с текущей сложностью. Такое решение будет неэффективным по памяти и по времени.

Задача 5. Звезды

I. Частичное решение, которое находит вертикальные или горизонтальные полоски. Точки находятся на одной прямой, если имеют одинаковые абсциссы или ординаты. И нужно найти прямую с наибольшим количеством точек. В отдельных массивах сохраняем абсциссы и ординаты. Сортируем массивы. Находим наиболее часто встречающиеся элементы: при проходе по массиву сравниваем текущий элемент с предыдущим, при равенстве элементов увеличиваем счётчик на единицу, при неравенстве — обновляем текущий максимум и присваиваем счетчику единицу.

II. Решение, неэффективное по времени. Основная идея решения: через две точки (*v* и *w*) можно провести прямую, третья точка (*t*) лежит на этой прямой, если у прямой, проходящей через точки *v* и *t*, тот же угол наклона.

Каждую точку v рассмотрим в паре с другими точками. При этом точки в паре будем всегда упорядочивать так, чтобы вторая точка по оси ординат была выше первой. Для каждой пары (*v*, *w*) проверим будет ли точка t, выбранная из оставшихся, лежать на той же прямой, что v и w. Для этого сравним тангенсы углов наклона к полуоси абсцисс *OX* прямых, проходящих через пары точек (*v*,*w*) и (*v*, *t*). Чтобы не переходить к действительным числам, тангенс не вычисляем напрямую, а храним в виде пары целых чисел — (числитель, знаменатель). При сравнении используем свойство пропорций: $(x_w-x_v)\cdot(y_t-y_v)=(x_t-x_v)\cdot(y_w-y_v)$, отдельно нужно рассматривать случай для равных ординат. Если точка *t* будет лежать на прямой с парой (*v*, *w*), то увеличиваем счётчик точек для пары (*v, w*). После проверки всех точек сравниваем счётчик для пары (*v, w*) с текущим максимумом для ответа.

III. Решение, эффективное по времени. Каждую точку *v* рассмотрим в паре с другими точками. При этом точки в паре будем всегда упорядочивать так, чтобы вторая точка по оси ординат была выше первой. Для прямой, проходящей через пару, находим и сохраняем в отдельном массиве тангенс угла наклона к полуоси абсцисс *OX*, тангенс храним в виде пары (числитель, знаменатель). Полученный массив сортируем и находим участок с наиболее часто встречающимся элементом. Он соответствует ответу для вершины *v*. Максимальный из ответов для всех вершин будет ответом на задачу.