

Задача А. Дорожная служба

В задаче необходимо выбрать минимальное число отрезков, накрывающих отрезков от 1 до 1000. Решается с помощью жадного алгоритма. Чтобы выбрать такое покрытие, нам необходимо отсортировать отрезки в порядке возрастания их начала. Посмотрим последовательно на все отрезки, которые накрывают 1 — начальную точку нужно покрыть обязательно, иначе не будет полного покрытия. Из всех этих отрезков лучше выбрать тот, который заканчивается позже остальных, потому что если выбрать его, а не другие отрезки, накрывающие 1, то покрытие всегда будет не хуже, а, вероятно, лучше, потому что этот отрезок покрывает больший участок пути. Далее необходимо покрыть его конец по такому же алгоритму: из всех отрезков, начинающихся не позже него (мы рассмотрим их всех последовательно), выберем тот, который закончится дальше всех. В ответе получается 11 отрезков. Один из правильных вариантов покрытия — отрезки 16, 17, 32, 42, 43, 71, 78, 80, 85, 98, 99. Задачу можно было решить, используя электронную таблицу: отсортировать данные по столбцу В, затем последовательно выбрать покрытие.

Задача В. Комната отдыха интровертов

В задаче необходимо выбрать максимальное число непересекающихся отрезков. Решается с помощью жадного алгоритма. Чтобы выбрать такой набор отрезков, нам необходимо отсортировать отрезки в порядке возрастания их концов. В качестве самого первого отрезка лучше всего выбрать тот, который заканчивается раньше. В таком случае набор отрезков будет не хуже, а, возможно, лучше, чем с любым другим отрезком, взятым в качестве первого. Далее необходимо рассмотреть отрезки, которые начинаются не раньше его конца. Самый первый такой отрезок нам подойдет как следующий — он заканчивается раньше остальных. С помощью такого алгоритма необходимо рассмотреть все отрезки. В ответе получается 53 отрезка.

Задача С. Звездное небо

В этой задаче необходимо рассмотреть события. Всего нас интересует два вида событий: когда звезда гаснет и когда снова загорается. Пометим первое событие как -1 , а второе событие, как 1 , а потом отсортируем их как пары координата и тип события по возрастанию. Изначально все звезды горят — потухших звезд 0 . Для каждого момента времени будем суммировать типы события с самого начала до этого момента — посчитаем баланс. Заметим, что значение погасших к моменту времени x звезд будет противоположно по значению балансу (он всегда будет не больше нуля). Нам необходимо узнать, когда баланс будет минимальным. Минимальный баланс в этой задаче -101 в моменты 840 и 849. Если одна звезда загорается, а другая гаснет, в задаче считаются как две погасшие звезды — так реализована ситуация, что два отрезка пересекаются в своих концах. Для того, чтобы обработать этот случай, гаснущие звезды помечены как -1 — в сортировке они будут раньше, чем загорающиеся, и баланс посчитается правильно. Задачу можно решить с помощью средств электронной таблицы: необходимо дописать в столбец В концы каждого отрезка, в столбце С первые 500 координат пометить как -1 , а следующие как 1 , отсортировать по столбцу С, затем по столбцу В, после чего сделать в столбце D префиксную сумму столбца С (записать в ячейке DX сумму элементов с C1 до CX включительно) — баланс, найти точки, в которых в столбце D получен минимум, это и будет ответом на задачу.

Задача D. Ровный танец

Заметим, что для того, чтобы найти минимальную по разнице между самым высоким и самым низким танцором в группе, танцоров необходимо отсортировать по росту и выбрать из всех подряд идущих десятков танцоров группу с минимальной разницей, из них выбрать ту, которая не ниже 175 см. Ответ — танцоры 274, 600, 654, 915, 83, 191, 215, 611, 750, 992.

Задача Е. Путешествие викинга

Давайте посчитаем лучший результат для каждой из клеток. Викинг движется только вправо — если он стоит на клетке i , то он не может вернуться назад и улучшить результат предыдущих клеток, поэтому мы можем использовать результат предыдущих для того, чтобы посчитать текущую.

Для первой клетки можно собрать 0 монет.

Для клетки с лесом оптимальный результат посчитать нельзя — она запрещена.

Для клетки с данью посмотрим, какие клетки для нее могли быть предыдущими.

Для второй клетки, если на ней нет леса, результат — число монет, которые нужно за нее заплатить, потому что мы пришли в нее с первой клетки.

Для всех клеток с третьей могло быть два варианта предыдущих: соседняя слева и через одну слева — выберем из них тот, который потребовал меньшей дани, к этому значению необходимо добавить количество монет, которое надо заплатить за текущую клетку. Мы считаем, что оптимальный путь — прийти в лучшую из предыдущих и перейти в текущую. Если одна из предыдущих была лесом, то мы выбираем единственный возможный вариант.

Таким образом, получается ответ 11 и 77, этот метод называется методом динамического программирования.