

Массивы в C++

Для решения задач необходимо зарегистрироваться на сайте

<https://informatics.msk.ru/>

Задачи указаны с номерами.

разбор

2768 Прыжки с трамплина

Задачу нужно решать в два этапа. На первом найти индексы «правильных» максимума и минимума, а потом вывести массив, заключив эти элементы в скобки.

Особенность этой задачи в том, чтобы найти именно первый минимум и последний максимум. Для этого просто поставим в условие разные знаки (< и >= соответственно).

Дальше нужно вывести массив почти стандартно. Когда доходим до соответствующих индексов, заключаем элементы в скобки.

1572 Уникальные элементы

Пройдем по массиву и для каждого элемента пройдем по массиву, считая количество равных ему. Если оно равно одному – выводить элемент.

3017 Последовательности

Технически сложная задача. Нужно сделать ровно то, что просят в условии. Будем несколько раз из одного массива «раскрывать» последовательность в другой, и перекидывать результат обратно в первый массив.

1461 Шарики

Ограничения задачи позволяют решать ее «в лоб».

Будем искать длинную цепочку и удалять ее – сдвигать конец массива влево, уменьшая размер.

Чтобы не обрабатывать конец последовательности отдельно, удобно поставить барьерный элемент.

3016 Списывание

По условию этой задаче списывать «по цепочке», то есть троечник не будет списывать у четверочника, который списывает у пятерочника. Поэтому нельзя делать изменения сразу в исходном массиве.

Заведем массив с первоначальными оценками и второй массив окончательных оценок.

Теперь можно просто смоделировать процесс, как написано в условии.

Пройдем по массиву и поставим во второй массив пятерки тем, кто сидят рядом с отличниками. А в первом массиве их оценки обнулим (они списывали и другим «не интересны»).

Снова пройдем по первому массиву и выставим всем, сидящим рядом с четверочниками четверки, а их оценки обнулим. Ту же операцию произведем с троечниками.

Таким образом, в первом массиве останутся оценки тех, кто не списывал и отличников, а во втором массиве тех, кто списывал.

В ответ вывести суммы соответствующих элементов двух массивов для каждого ученика.

Нужно либо корректно обрабатывать границы дополнительными условиями, либо использовать барьерные элементы, чтобы у первого и последнего ученика появились соседи. Например, чтобы это были «ученики» с оценкой 0. При этом надо завести оба массива размером $N + 2$.

111160 Обувной магазин

Отсортируем обувь по неубыванию размера и применим жадный алгоритм: как только размер очередной пары больше последней "надетой" хотя бы на 3 размера, увеличиваем счетчик количества пар и меняем последний использованный размер. В качестве первой пары нам подходит минимальная из тех, которые больше или равны размеру ноги покупателя.

734 Такси

Нужно посадить самых «далекоживущих» в самое дешевое такси. Но ведь требуется вывести номера такси. Заведем еще два массива. Массив номеров такси и массив номеров пассажиров (заполним их последовательно). Отсортируем массив тарифов такси по возрастанию, а массив расстояний по убыванию. Нельзя ни в коем случае сортировать просто исходные массивы отдельно – номера «оторвутся» от расстояний. Когда мы в сортировке меняем элементы массива расстояний, тут же, синхронно, надо менять и элементы массива номеров сотрудников. В

конце просто выведем массив номеров такси, предварительно отсортируем массив номеров сотрудников по возрастанию.

Можно использовать любую квадратичную сортировку.

3594 Злые свинки или Anti Angry Birds

Нужно сбить все вертикали, а это значит, что сколько и каких птиц, с какими ординатами (игреками) на каждой вертикали не важно. Можно даже не читать массив «игреков», а ограничиться чтением массива «иксов». Дальше нужно узнать, сколько различных чисел в этом массиве. Для этого отсортируем массив. Пройдемся по нему и посчитаем количество различных элементов.

113612 Подборка новостей

Решение за $O(n^2)$

Перебираем индекс начала отрезка i . Для каждого начала переберем конец отрезка j такой, что $i \leq j < i+n$, при этом нужно брать индексы по модулю n , т. к. массив зацикленный (или же можно прибавить массив a сам к себе). В процессе перебора j считаем сумму на отрезке от i до j . Так будут рассмотрены все возможные отрезки, среди них выберем искомый.

*Решение за $O(n * \log(n))$*

Будем перебирать индекс начала отрезка i , и для каждого i бинарным поиском искать минимальный конец отрезка j такой, что сумма $a_i + a_{i+1} + \dots + a_j \geq p$. Чтобы находить сумму на отрезке за $O(1)$, предподсчитаем префиксные суммы для массива a . Среди всех найденных i, j выберем оптимальную.

Решение за $O(n)$

Применим метод двух указателей. Пусть изначально отрезок начинается в первом элементе массива, то есть $i = 0$. Найдем минимальный конец подходящего отрезка j . Сдвинем начало отрезка от i к $i+1$: сумма на отрезке уменьшилась на a_i . Затем увеличим j на такое минимальное $k \geq 0$, чтобы сумма на отрезке $[i+1; j+k]$ была не меньше p . При увеличении j будем пересчитывать сумму на текущем отрезке. Среди всех найденных i, j выберем оптимальную.