

Задание 5

На прямой расположены n стойл (даны их координаты на прямой), в которые необходимо расставить k коров так, чтобы минимальное расстояние между коровами было как можно больше. Гарантируется, что $1 < k < n$

входные данные

6 3

2 5 7 11 15 20

выходные данные

9

Задание 6

Есть два принтера, один печатает лист раз в x минут, другой раз в y минут. За сколько минут они напечатают N листов? $N > 0$. Нельзя решать формулой.

Задание 7

Имеется отсортированный массив из n целых чисел, который был циклически сдвинут неизвестное число раз. Напишите код для поиска элемента в массиве.

Предполагается, что массив изначально был отсортирован по возрастанию.

Пример:

Ввод: найти 5 в {15, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14}.

Вывод: 8 (индекс числа 5 в массиве).

Задание 16

Даны два целочисленных массива A и B размера m и n , соответственно. Необходимо найти пересечение этих массивов. Числа в пересечении могут располагаться в любом порядке.

Пример:

Input: $A[] = \{1,4,3,2,5, 8,9\}$, $B[] = \{6,3,2,7,5\}$

Output: {3,2,5}

Input : $A[] = \{3,4,6,7,10, 12, 5\}$, $B[] = \{7,11,15, 18\}$

Output: {7}

Задание 17

У вас есть n монет и вы хотите выстроить их в форме лестницы, где на k -ом ряду будет ровно k монет. Найдите количество полных рядов, которые будут сформированы. n - неотрицательное число в пределах $\text{int}32$.

Задание 18

Предложите алгоритм, который ищет значение (true/false) в матрице $m \times n$

- Числа в каждом ряду отсортированы по возрастанию
- Первое число каждого ряда больше, чем последнее число предыдущего ряда

Пример

1	3	5	7
10	11	16	20
23	30	34	60

Задание 19

Дана матрица $n \times n$, где столбцы и строки отсортированы по возрастанию. Необходимо найти k -й минимальный элемент матрицы.

Пример:

```
matrix = [  
  [ 1, 5, 9],  
  [10, 11, 13],  
  [12, 13, 15]  
],  
k = 8,  
  
return 13.
```

Задание 20

Даны два отсортированных массива. Требуется найти их медиану.

A  m=6

B  n=5

$A \cup B$  m+n=11

$A \cup B$  m+n=11

Задание 1

Дан массив из $2n$ элементов. Требуется разбить его на n пар $(a_1, b_1), (a_2, b_2) \dots$ так, чтобы сумма $\min(a_i, b_i)$ была максимальной.

Пример

Ввод: [1, 4, 3, 2]

Вывод: 4

Объяснение, $n = 2$; $\min(3, 4) + \min(1, 2) = 3 + 1 = 4$

Задание 2

Дан массив неотрицательных целых чисел. Необходимо найти максимально возможный периметр треугольника, который можно сформировать, используя длины, имеющиеся в массиве. В случае, если это невозможно, выведите -1

Пример

Ввод: [6, 4, 2, 4, 1, 6, 7, 4, 8, 3, 5]

Вывод: 21

Ввод: [2, 20, 7, 55, 1, 33, 12, 4]

Вывод: -1

Задание 3

Дан массив из n целых чисел. Требуется определить минимальную абсолютную разность X между элементами массива и вывести в порядке возрастания все пары элементов, разность между которыми равна X

Пример

Ввод: [1, 4, 3, 2]

Вывод: [[1, 2], [2, 3], [3, 4]]

Задание 4

Дан массив из n целых чисел. Требуется вывести все тройки чисел a , b и c такие, что $a + b + c = 0$,

Пример

Ввод: [-1, 0, 1, 2, -1, -4]

Вывод: [[-1, 0, 1], [-1, -1, 2]]

Задание 5

Дан связный список. Необходимо отсортировать его, используя сортировку вставками.

Пример

Ввод: 8->0->1->3

Вывод: 0->1->3->8

1. В последовательности записаны целые числа от 1 до N в произвольном порядке, но одно из чисел пропущено (остальные встречаются ровно по одному разу). N заранее неизвестно. Определить пропущенное число.
2. В последовательности записаны целые числа. Одно из чисел встречается ровно один раз, остальные — по два раза. Найти число, которое встречается один раз.
3. Дан массив целых чисел. Требуется найти равновесный индекс, то есть такой индекс, что сумма всех элементов с меньшими индексами равняется сумме элементов с большими индексами, или вывести -1 если такое невозможно.
Пример: -7, 1, 5, 2, -4, 3, 0. Ответ: 3, потому что $-7+1+5 = -4+3 = -1$
4. Дан массив целых чисел, необходимо определить, существует ли в данном массиве подмассив с нулевой суммой элементов.

Примеры:

- [4, 2, -3, 1, 6] -> да, $2-3+1=0$
 - [4, 2, 0, 1, 6] -> да, [0] является подмассивом с нулевой суммой элементов
 - [-3, 2, 3, 1, 6] -> нет
5. Дана последовательность натуральных чисел. Необходимо найти максимально возможную сумму её непрерывной подпоследовательности, в которой количество чётных элементов кратно $k = 10$.
 6. В длинной очереди стоят люди. Для каждого из них, кроме последнего, записали его имя и имя того, кто стоит за ним. Полученные записи перемешали и записали в файл. Требуется за один просмотр файла определить имена первого и последнего человека. Известно, что эти имена различны (иначе задача неразрешима), но, в общем, имена могут повторяться. Имя каждого человека состоит из шестнадцати 8-битных символов.
 7. В последовательности записаны целые числа, больше половины из которых равны одному и тому же числу X . За один просмотр последовательности найти это число.
 8. В последовательности записаны целые неотрицательные числа, меньшие M , причём известно, что каждое число встречается не более одного раза. Найти наименьшее число, которое в этой последовательности не встречается.
 9. В последовательности записано $M+1$ целое неотрицательное число, все числа меньше M . Найти какое-нибудь число, которое встречается хотя бы дважды.

12. Дан массив целых чисел, необходимо найти k -й максимальный элемент в массиве.

Пример: $arr = [7, 4, 5, 3, 9, 1]$; $k = 2$ -> $res = 7$

13. На вход программе последовательно подаются целые числа. Необходимо найти медиану введенных на данный момент чисел. Дубликатов нет.

Пример: 5, 15, 1, 3 -> Вывод: 5, 10, 5, 4

14. Даны k отсортированных по возрастанию связных списков, необходимо объединить их в один отсортированный по возрастанию список.

Пример:

1->5->7

2->3->6->9

4->8->10

Вывод: 1->2->3->4->5->6->7->8->9->10

10. Дан набор из n отрезков на прямой, заданных координатами начал и концов $[l_i, r_i]$. Требуется найти любую точку на прямой, покрытую наибольшим количеством отрезков.

11. Дан набор из n отрезков на прямой, заданных координатами начал и концов $[l_i, r_i]$. Требуется найти суммарную длину их объединения.

12. Даны q точек. Надо для каждой точки определить, скольким отрезкам (из задач выше) принадлежит данная точка.

Задание 5

На прямой расположены n стойл (даны их координаты на прямой), в которые необходимо расставить k коров так, чтобы минимальное расстояние между коровами было как можно больше. Гарантируется, что $1 < k < n$

=====

ОТВЕТ:

https://wiki.algocode.ru/index.php?title=Бинарный_поиск_по_ответу

<https://ru.algorithmica.org/cs/interactive/answer-search/> более аккуратное решение

Обозначим искомое минимальное расстояние за X и делаем бинарный поиск по ответу, где минимум - 0, а максимум - разность координат самого ближнего и самого дальнего стойла + 1.

входные данные

6 3

2 5 7 11 15 20

выходные данные

9

Код решения:

```
bool check(int x) {
    int cows = 1;
    int last_cow = coords[0];
    for (int c : coords) {
        if (c - last_cow >= x) {
            cows++;
            last_cow = c;
        }
    }
    return cows >= k;
}
```

```
int solve() {
    sort(coords.begin(), coords.end());
    int l = 0; // так как коров меньше, чем стойл, x = 0 нам всегда хватит
    // по условию есть хотя бы 2 коровы,
    // которых мы в лучшем случае отправим в противоположные стойла:
    int r = coords.back() - coords[0] + 1;
    while (r - l > 1) {
        int m = (l + r) / 2;
        if (check(m))
            l = m;
        else
            r = m;
    }
    return l;
}
```

разбор примера из условия

$$n = 6$$

$$k = 3$$

0	1	2	3	4	5
2	5	7	11	15	20

1 шаг

$$l_{\text{mid}} = 0$$

$$r_{\text{mid}} = 20 - 2 + 1 = 19$$

$x = 19$, проверим можно ли всех коров расставить на расстоянии 19, это функция `check`
false

2 шаг

$m = 10$ - это x , наше расстояние между коровами, расставить нельзя
false 2 корова

$$r = 10, l = 0$$

$\frac{r}{l}$	$\frac{r}{l}$	$\frac{r}{l}$	$\frac{r}{l}$	$\frac{r}{l}$	$\frac{r}{l}$
✓	-	-	-	✓	-
2	5	7	11	15	20

3 шаг

$m = 5$, расставить еще нельзя потому что 3 коровы, можно

✓	-	✓	-	✓	-
2	5	7	11	15	20

можно увеличить расстояние!
 $r = 10, l = 5$

4 шаг

$m = 7$, опять расставим

✓	-	-	✓	-	✓
2	5	7	11	15	20

$r = 10, l = 7$

5 шаг

$m = 8$, расставим

✓			✓		✓
2	5	7	11	15	20

$r = 10, l = 8$

6 шаг

$m = 9$, расставим

✓			✓		✓
2	5	7	11	15	20

$r = 10, l = 9$, выход

$n = 6$
 $k = 3$

0	1
2	5

1 шаг
 $l_{\text{ung}} =$
 $r_{\text{ung}} =$
 $x = 19,$

2 шаг

$m =$

$r = 1$

3 шаг

$m =$

$m =$

$r =$

Задание 6

Есть два принтера, один печатает лист раз в x минут, другой раз в y минут. За сколько минут они напечатают N листов? $N > 0$. Нельзя решать формулой.

=====

ОТВЕТ:

https://wiki.algocode.ru/index.php?title=Бинарный_поиск_по_ответу

Берем границы, например, 0 и Nx (потому что два принтера напечатают N листов не медленнее чем один принтер)

=====

Задание 7

Имеется отсортированный массив из n целых чисел, который был циклически сдвинут неизвестное число раз. Напишите код для поиска элемента в массиве.

Предполагается, что массив изначально был отсортирован по возрастанию.

Пример:

Ввод: найти 5 в {15, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14}.

Вывод: 8 (индекс числа 5 в массиве).

=====

ОТВЕТ:

<https://tproger.ru/problems/searching-element-in-array/>

Идея: Массив после сдвига состоит из двух отсортированных кусочков. Сначала бинарным поиском можно найти “точку разрыва” (находим середину и смотрим, левому или правому массиву она принадлежит), а потом в нужном подмассиве ищем элемент.

В случае равных элементов придется проверять оба подмассива, поэтому логарифмическая сложность превратится в линейную.

Задание 16

Даны два целочисленных массива A и B размера m и n , соответственно. Необходимо найти пересечение этих массивов. Числа в пересечении могут располагаться в любом порядке.

Пример:

Input: $A[] = \{1,4,3,2,5, 8,9\}$, $B[] = \{6,3,2,7,5\}$

Output: {3,2,5}

Input : $A[] = \{3,4,6,7,10, 12, 5\}$, $B[] = \{7,11,15, 18\}$

Output: {7}

=====

ОТВЕТ

<https://afteracademy.com/blog/find-the-intersection-of-two-unsorted-arrays>

Рассмотреть надо первые два решения:

1. Brute force, перебираем все элементы за $O(m*n)$
2. Сортируем массив B и для каждого элемента из A делаем бинарный поиск за $O(n*\log(n) + m*\log(n))$
3. Сортируем оба массива и идем двумя указателями за $O(n*\log(n) + m*\log(m))$
4. Делаем хэш-таблицу из массива B и ищем в ней каждый элемент из A за $O(n+m)$

Approach	Time Complexity	Space Complexity
Using Nested Loops	$O(m*n)$	$O(1)$
Sorting and binary search	$O(n\log n + m\log n)$	$O(1)$ (Heap Sort) $O(n)$ (Merge Sort)
Sorting and two-pointer approach	$O(m\log m + n\log n)$	$O(1)$ (Heap Sort) $O(n)$ (Merge Sort)
Hash Table	$O(n+m)$	$O(n)$

original

Problem Description: Given two integer arrays $A[]$ and $B[]$ of size m and n , respectively. We need to find the intersection of these two arrays. The intersection of two arrays is a list of distinct numbers which are present in both the arrays. The numbers in the intersection can be in any order.

=====

Задание 17

У вас есть n монет и вы хотите выстроить их в форме лестницы, где на k -ом ряду будет ровно k монет. Найдите количество полных рядов, которые будут сформированы. n - неотрицательное число в пределах int32 .

=====

ОТВЕТ

<https://just4once.gitbooks.io/leetcode-notes/content/leetcode/binary-search/441-arranging-coins.html>

Бинарный поиск по ответу, на основе того что $x(x+1) / 2 \leq n$, а максимальное значение $x = n$

original

You have a total of n coins that you want to form in a staircase shape, where every k -th row must have exactly k coins.

*Given n , find the total number of **full** staircase rows that can be formed.*

n is a non-negative integer and fits within the range of a 32-bit signed integer.

=====

Задание 18

Предложите алгоритм, который ищет значение (true/false) в матрице $m \times n$

- Числа в каждом ряду отсортированы по возрастанию
- Первое число каждого ряда больше, чем последнее число предыдущего ряда

Пример

1	3	5	7
10	11	16	20
23	30	34	60

=====

ОТВЕТ

<https://medium.com/theleanprogrammer/74-search-a-2d-matrix-5383443563a1>

Плохое решение - делать бинпоиск по каждому ряду за $O(m \cdot \log(n))$

Хорошее решение - делать бинпоиск по всей матрице сразу за $O(\log(nm))$

original

Write an efficient algorithm that searches for a value in a $m \times n$ matrix.

This matrix has the following properties:

- Integers in each row are sorted from left to right.
- The first integer of each row is greater than the last integer of the previous row.

Example 1:

1	3	5	7
10	11	16	20
23	30	34	60

```
Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3
Output: true
```

=====

Задание 19

Дана матрица $n \times n$, где столбцы и строки отсортированы по возрастанию. Необходимо найти k -й минимальный элемент матрицы.

Пример:

```
matrix = [  
  [ 1, 5, 9],  
  [10, 11, 13],  
  [12, 13, 15]  
],  
k = 8,  
  
return 13.
```

=====

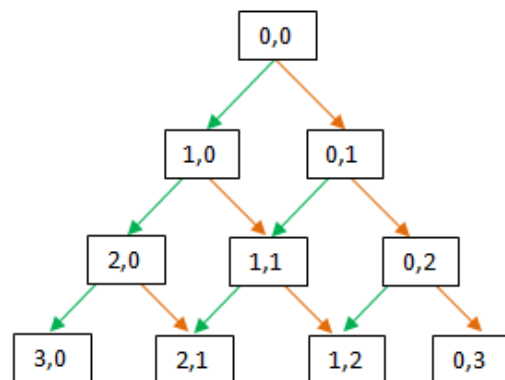
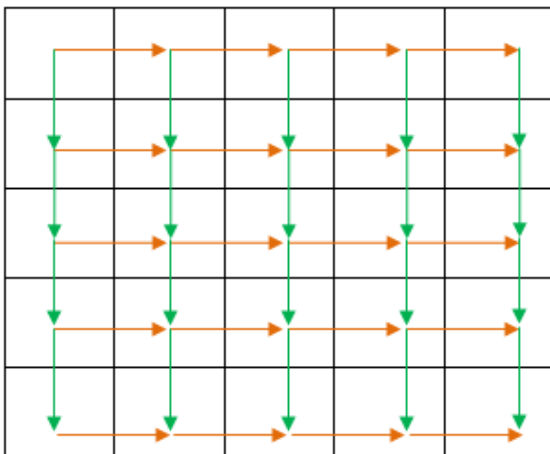
ОТВЕТ

<https://ayoubomari.medium.com/kth-smallest-element-in-sorted-matrix-b20400cf878e>

Возможные решения:

1. Brute force за $O(N^2 \cdot \log(N^2))$ - объединить в массив, посортировать, взять элемент k
2. Воспользуемся сортированностью строк и будем мерджить - если делать последовательно, то $O(N^3)$, если "пирамидкой", то $O(N^2 \cdot \log(N))$, что хуже чем предыдущий метод
3. Если использовать очередь с приоритетами, то работает за $O(k \cdot \log(k))$

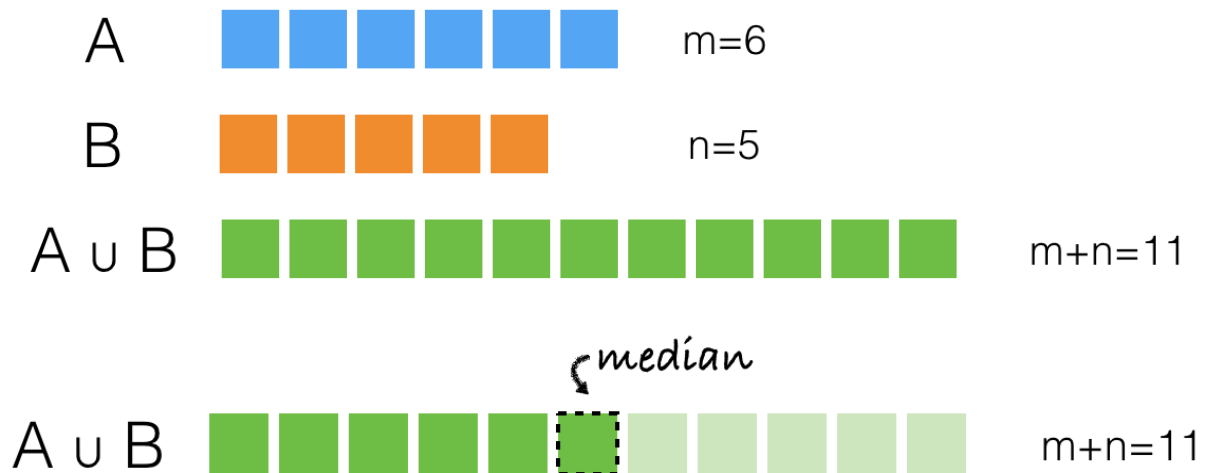
Почему можно использовать очередь с приоритетами -



=====

Задание 20

Даны два отсортированных массива. Требуется найти их медиану.



=====

ОТВЕТ

<https://medium.com/@hazemu/finding-the-median-of-2-sorted-arrays-in-logarithmic-time-1d3f2ecbeb46>

(Оригинал на литкоде <https://leetcode.com/problems/median-of-two-sorted-arrays/>)

Идея: для поиска медианы надо знать только элементы до этой самой медианы (первую половину объединенного массива), а для этого можно определить бинарным поиском размер вклада массива A в эту половину.

=====

Sources:

1. <https://lenchen.medium.com/leetcode-561-array-partition-i-71541575b3ff>
2. <https://www.geeksforgeeks.org/maximum-perimeter-triangle-from-array/>
3. <https://afteracademy.com/blog/minimum-absolute-difference-in-an-array>
4. <https://fizzbuzzed.com/top-interview-questions-1/>
5. <https://afteracademy.com/blog/sort-a-linked-list-using-insertion-sort>

1. В последовательности записаны целые числа от 1 до N в произвольном порядке, но одно из чисел пропущено (остальные встречаются ровно по одному разу). N заранее неизвестно. Определить пропущенное число.

Решение - посчитать сумму S и количество K, пропущенное число - $(K+1)(K+2)/2 - S$

2. В последовательности записаны целые числа. Одно из чисел встречается ровно один раз, остальные — по два раза. Найти число, которое встречается один раз.

Пройтись по всем числам и найти их XOR. *Можно подумать про аналогичные задачи, где X встречается 1-2 раза, а остальные числа 3 раза (решение в трюковой системе счисления), а также где X встречается 1-3*

раза, а остальные числа 4 раза. Ответы - задачи 3 и 4
<https://habr.com/ru/post/243819/>

3. Дан массив целых чисел. Требуется найти равновесный индекс, то есть такой индекс, что сумма всех элементов с меньшими индексами равняется сумме элементов с большими индексами, или вывести -1 если такое невозможно. Пример: -7, 1, 5, 2, -4, 3, 0. Ответ: 3, потому что $-7+1+5 = -4+3 = -1$

Идеи решения

- Перебирать все индексы и считать суммы слева и справа - квадрат**
- Найти сумму всех элементов, а вторым проходом считать сумму элементов слева от текущего и сравнивать - линия**
- Посчитать префиксную сумму слева и справа, а потом найти точку равенства - линия**

<https://www.geeksforgeeks.org/equilibrium-index-of-an-array/>

4. Дан массив целых чисел, необходимо определить, существует ли в данном массиве подмассив с нулевой суммой элементов.

Примеры:

- [4, 2, -3, 1, 6] -> да, $2-3+1=0$
- [4, 2, 0, 1, 6] -> да, [0] является подмассивом с нулевой суммой элементов
- [-3, 2, 3, 1, 6] -> нет

Решения:

- **Перебор всех подмассивов за квадрат**
- **Подсчет префиксной суммы для всех элементов и складывание их в сет. Как только префиксная сумма повторяется, значит, есть подмассив нулевой длины.**
- <https://www.geeksforgeeks.org/find-if-there-is-a-subarray-with-0-sum/>

5. Дана последовательность натуральных чисел. Необходимо найти максимально возможную сумму её непрерывной подпоследовательности, в которой количество чётных элементов кратно $k = 10$.

Будем последовательно считывать числа из файла. В массив lefts будем записывать первые встречающиеся суммы с количеством чётных элементов, делящимся на 10 с остатком от нуля до девяти. В массив rights также будем записывать суммы с количеством чётных элементов, делящимся на 10 с остатком от нуля до девяти. Если будет встречено несколько сумм с одним и тем же остатком, в массив rights будет записана сумма, встретившаяся позже. Если после считывания всех чисел из файла значение в переменной count не кратно 10, тогда будем проверять разности элементов массивов rights и lefts с соответствующими индексами и выводить на экран наибольшую из таких разностей — это и будет искомой максимальной суммой.

<https://inf-ege.sdamgia.ru/problem?id=38961>

6. В длинной очереди стоят люди. Для каждого из них, кроме последнего, записали его имя и имя того, кто стоит за ним. Полученные записи перемешали и записали в файл. Требуется за один просмотр файла определить имена

первого и последнего человека. Известно, что эти имена различны (иначе задача неразрешима), но, в общем, имена могут повторяться. Имя каждого человека состоит из шестнадцати 8-битных символов.

7. В последовательности записаны целые числа, больше половины из которых равны одному и тому же числу X . За один просмотр последовательности найти это число.
8. В последовательности записаны целые неотрицательные числа, меньше M , причём известно, что каждое число встречается не более одного раза. Найти наименьшее число, которое в этой последовательности не встречается.
9. В последовательности записано $M+1$ целое неотрицательное число, все числа меньше M . Найти какое-нибудь число, которое встречается хотя бы дважды.

Решения последних четырех задач - <https://habr.com/ru/post/243819/>

1. Дан массив целых чисел, необходимо найти k -й максимальный элемент в массиве.
Пример: $arr = [7, 4, 5, 3, 9, 1]$; $k = 2 \rightarrow res = 7$
 1. **вар. 1. Строим max-heap и удаляем $(k-1)$ первых элементов. k -й наибольший будет в корне.**
 2. **Вар. 2. Строим min-heap на k элементов, а все остальные элементы добавляем в кучу, заменяя корень (если элемент больше корня). Так в куче останется k самых больших элементов и k -й будет в корне.**
 3. <https://www.techiedelight.com/find-kth-largest-element-array/>
2. На вход программе последовательно подаются целые числа. Необходимо найти медиану введенных на данный момент чисел. Дубликатов нет.
Пример: 5, 15, 1, 3 \rightarrow Вывод: 5, 10, 5, 4
 1. **Сортировка вставками - квадрат**
 2. **Две кучи - max-heap для элементов слева от медианы и min-heap для элементов справа от медианы. Когда общее число элементов четно, берем среднее двух корней, когда нечетно - берем корень большей кучи**<https://www.geeksforgeeks.org/median-of-stream-of-integers-running-integers/>
3. Даны k отсортированных по возрастанию связных списков, необходимо объединить их в один отсортированный по возрастанию список.
Пример:
1 \rightarrow 5 \rightarrow 7
2 \rightarrow 3 \rightarrow 6 \rightarrow 9
4 \rightarrow 8 \rightarrow 10
Вывод: 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10
 1. **Если $k=2$, то сравниваем два корня и объединяем за $O(m+n)$, где m и n - длины списков**
 2. **Для $k \geq 3$ можно объединить все списки в один друг за другом, а потом сделать merge sort за $O(N \log N)$, где N - суммарное число элементов**

3. Строим min-heap из первых элементов списков. Затем извлекаем корень из кучи и добавляем в нее элемент из того списка, из которого был извлеченный элемент, до тех пор, пока не кончатся элементы. $O(N \log k)$ по времени и $O(k)$ по памяти
4. Разделяй и властвуй - объединяем два списка в 1, еще два списка в 1, потом эти объединенные списки в 1 и так далее. Слияние двух списков - $O(N)$, количество слияний - $O(\log k)$ -> $O(N \log k)$ по времени и $O(1)$ дополнительной памяти

<https://www.techiedelight.com/efficiently-merge-k-sorted-linked-lists/>

1. Дан набор из n отрезков на прямой, заданных координатами начал и концов $[l_i, r_i]$. Требуется найти любую точку на прямой, покрытую наибольшим количеством отрезков.

Имеет смысл рассматривать только "интересные" точки - те, в которых начинается или заканчивается отрезок - и считать количество отрезков, их покрывающих. Простой перебор - n^2 , использование сортировки - $n \log n$, надо обратить внимание на точки, где и начинаются, и заканчиваются отрезки - должны быть учтены и те и те.

<https://ru.algorithmica.org/cs/decomposition/scanline/>

2. Дан набор из n отрезков на прямой, заданных координатами начал и концов $[l_i, r_i]$. Требуется найти суммарную длину их объединения.

Аналогично предыдущей задаче, считаем длину отрезка между интересными точками, прерывая объединение если число отрезков, покрывающих точку, стало равным нулю.

3. Даны q точек. Надо для каждой точки определить, скольким отрезкам (из задач выше) принадлежит данная точка.

Считаем все точки и почитаем их фиктивным отрезком (не добавляющимся в сумму отрезков). Потом сделаем сортировку все отрезков и получим решение за $O((n+q) \log(n+q))$