1. **Задача 1**

   Дан массив целых чисел и число target.
   Нужно найти непустой подотрезок (непрерывную подпоследовательность) с заданной суммой target, либо сказать, что это невозможно.
   Если такие отрезки есть, то надо вернуть любой отрезок (как индексы концов, включительно).
   Если нет, то (-1,-1)
2. **Задача 2**

   Дан список целых чисел, повторяющихся элементов в списке нет.
   Нужно преобразовать это множество в строку, сворачивая соседние по значению числа в диапазоны.

   Примеры:
   [1,4,5,2,3,9,8,11,0] => "0-5,8-9,11"
   [1,4,3,2] => "1-4"
   [1,4] => "1,4"
   [1,2] => "1-2"
3. **Задача 3**

   Дано бинарное дерево с выделенным корнем, в каждой вершине которого записано по одной букве A-Z.
   Две вершины считаются эквивалентными, если поддеревья этих вершин содержат одинаковое множество (т.е. без учета частот) букв.
   Нужно найти две эквивалентные вершины с максимальным суммарным размером поддеревьев.
4. **Задача 4**

   Дан массив точек с целочисленными координатами (x, y). Определить, существует ли
   вертикальная прямая, делящая все точки, не лежащие на ней, на 2 симметричных
   относительно этой прямой множества.
5. **Задача 5**

   Дан массив каждый элемент массива это два числа первое число это день заезда второе день выезда нужно найти количество элементов у которых есть пересечения причем если одна пара 1-5 а вторая 5-1 то это не считается пересечением а считается пересечением 1-5 и допустим 2-18
6. **Задача 6 (Из Яндекса)**

   Дана строка, состоящая из букв A-Z:
   * "AAAABBBCCXYZDDDDEEEFFFAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBB"
   * Нужно написать функцию RLE, которая на выходе даст строку вида:
   * "A4B3C2XYZD4E3F3A6B28"
   * И сгенерирует любую ошибку, если на вход пришла невалидная строка.
   *
   * Пояснения:

* 1. Если символ встречается 1 раз, он остается без изменений

* 2. Если символ повторяется более 1 раза, к нему добавляется количество повторений

7. **Задача 7**

a = 5

b = 10

без циклов и условных операторов не вводя третью переменную поменять местами a и b т.е. a = 10 b = 5;

8. **Задача 8**

```
/TODO:
public class SortByCategoryWithEquidistanceFormalization {


 /**
  * отсортировать лист таким образом, что бы все элементы были равнораспределены по результирующему листу, и при повторном запуске цикла результат не менялся
  * <p>
  * Выборка категории 1,1,1,2,2,3 Неправильно: категория 1, 1 .... т.к. две идут подряд Неправильно: категория 1, 2, 1, .... т.к. 1 повторилась до того как вставлено 3
  * <p>
  * Правильно н-р 1, 2, 3, 1, 2, 1
  */
 public static void main(String[] args) {

  var data = List.of(
        new Product("продукт 1", "категория 1"),
        new Product("продукт 2", "категория 1"),
        new Product("продукт 3", "категория 1"),
        new Product("продукт 4", "категория 2"),
        new Product("продукт 5", "категория 2"),
        new Product("продукт 6", "категория 3")
  );
  //System.out.println(sortedData); //TODO
 }
 /**
  * Можно менять
  */
 static class Product {

  private final String value;
  private final String category;
```

```
    public Product(String value, String category) {
      this.value = value;
      this.category = category;
    }
  }
}
```

9. **Задача 9**

   Из листа, хранящих Person, создать мапу, где ключ - язык Person (у Person есть поле language), а значение - коллекция Person с этим языком.
   Язык у каждого Person только один, это нативный язык.

10. **Задача 10 (Это очень популярный тип задач на собесах)**

```
// Напишите программу на Java для подсчета количества
// конкретных слов в строке, используя HashMap.
import java.io.*;
import java.util.*;

class MyCode {
  public static void main (String[] args) {
}
```

11. **Задача 11**

```
Необходимо outputSize раз взять случайным образом элементы из списка input и поместить
        * их в выходной список. Повторения допустимы.
        *
        * @param input входной список
        * @param outputSize размер выходного списка
        * @return выходной список со случайными элементами из входного списка
        */
abstract List<String> randomizeList(List<String> input, int outputSize);
```

12. ***Задача 12***

    Top N problem
    Description
    You are tasked with implementing a class that handles price updates from the market and can output N highest unique price values for the whole class runtime.

    N is immutable throughout the application runtime and should be defined on startup. (Проблема номер N Описание Вам поручено реализовать класс, который обрабатывает обновления цен с рынка и может выводить N самых высоких

уникальных значений цен за все время работы класса. N неизменяем во время выполнения приложения и должен определяться при запуске.)
Your class must have the following methods:
void push(int val);
Collection<Integer> top();
Where push is called for each new price value and top is called when result is needed. Assume that this is a single threaded environment.
Input
Infinite stream of non-unique integer values will be fed into handler class by calling push method
Output
When top method is called you should return a Collection of N highest unique price values received by your class throughout whole application runtime. Order of the values in a resulting collection doesn't matter
(Ваш класс должен иметь следующие методы: void push (int val); Collection<Integer> top(); Где push вызывается для каждого нового значения цены, а top вызывается, когда необходим результат. Предположим, что это однопоточная среда. Вход Бесконечный поток неуникальных целочисленных значений будет передан в класс обработчика путем вызова метода push. Выход При вызове метода top вы должны вернуть коллекцию из N самых высоких уникальных значений цен, полученных вашим классом за все время выполнения приложения. Порядок значений в результирующей коллекции не имеет значения.)

13. **Задача 13**

CITY
id
code
name

DISTRICT
id
city_id
name

HOUSE
id
district_id
number_of_floors

SQL задача.
Вывести наименование районов, где code города = мск, где есть дома с кол-вом этажем менее 6
И число таких домов на районе более одного

14. **Задача 14**

Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must be unique and you may return the result in any order.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
Explanation: [4,9] is also accepted.

15. **Задача 15 (Популярная задача)**

- Алгоритмическая задача . Написать метод, который должен принимать n количество объектов(строк), в методе вернуть строку с содержанием этих объектов вставив между ними разделитель "-";

16. **Задача 16 (Популярная задача)**

Алгоритмическая задача: реализовать метод который на вход принимает 2 строки и проверят, являются ли они анаграммами.

17. ***Задача 17***

//Задача: найти первый не повторяющийся элемент в наборе целых чисел
//9, 4, 9, 9, 6, 7, 4, 5, 5 -> 6
//5, 9, 8, 5, 7, 9, 8, 7, 1 -> 1

18. ***Задача 18***

```
//  Найти и вывести в консоль все не повторяющиеся элементы в массиве
//  input [9, 4, 9, 6, 7, 4, 5]
//  что должно быть напечатано в консоли [6, 7, 5]
//


 public class Solution {
    public static void main(String[] args) {
      int[] arr = {9, 4, 9, 6, 7, 4, 5, 9, 9};
 }
```

19. ***Задача 19***

// Вернуть все дубликаты которые встречаются среди первого или второго массива.

// Дубликатами считаются числа, которые находятся либо в одинаковых, либо в разных массивах.

// Дубликаты должны быть возвращены в том же порядке в котором они встречаются сначала в первом массиве, затем во втором

//

// Например:
// a = [4, 4, 3, 9, 8]
// b = [2, 1, 2, 9]
//
// Ожидаемый результат:
// [4, 4, 9, 2, 2, 9]

20. *Задача 20*

- Алгоритмическая задача : Вывести все дубликаты, оценить асимптотического сложность (Если Вы скажете вариант с ХешМапой то сложность будет n log2n

21. Задача 21

- Задача по стримам, вывести по условию. Как использовать метод фильтр без лямбды

Написать по сути пул интов. Конструктор уже реализован, в него дополнять не надо, остальное дописать.

java

```
public class Integer {
   public Integer(int i){
      // makes a new wrapper
   }
   // ========================


   public static Integer valueOf(int i){
      // [-128,127]
      // to implement

   }

   // Tests
   // valueOf(1) == valueOf(1)
   // valueOf(128) != valueOf(128)
}
```

22. Задача 22

Матрица должна быть валидна в соответствии с нижележащими правилами
1) Каждая строка должна содержать цифру от 1 до 9 без повторений
2) Каждая колонка должна содержать цифру от 1 до 9 без повторений
3) Матрица может частично заполнена, пустые ячейки содержат '.'
4) Значение ячейки валидно если это цифра от 1 до 9 или .
4) Наличие букв в качестве значений недопустимо
**/*

```
public class Main {
```

```
    public static void main(String[] args) {
        char[][] matrix =
                {{'5','3','.','.','7','.','.','.','.'}
                        ,{'6','.','.','1','9','5','.','.','.'}
                        ,{'.','9','8','.','.','.','.','.','6','.'}
                        ,{'8','.','.','.','.','6','.','.','.','3'}
                        ,{'4','.','.','8','.','3','.','.','1'}
                        ,{'7','.','.','.','.','2','.','.','.','.','6'}
                        ,{'.','6','.','.','.','.','.','.','2','8','.'}
                        ,{'.','.','.','.','4','1','9','.','.','5'}
                        ,{'.','.','.','.','.','8','.','.','7','9'}};

    }

    public boolean isValidMatrix(char[][] matrix) {

    }

}
```

23. Задача 23

- Устная алгоритмическая задача. Дан массив интов в диапазоне [1,100], как найти отсутствующее число

24. Задача 24

Попалась задачка: написать метод, который принимает лист интов и который считает количество дубликатов чисел. Результат вывести в виде {число -> количество дублей}.

25. Задача 25

Напишите функцию, которая для строки возвращает первый символ, встречающийся в строке только один
раз.
Примеры входа и выхода функции:
Вход -> Выход
abacaba -> c
melon -> m
extends -> x

26. Задача 26

Напишите функцию, которая проверяет строку, состоящую из символов "(",")", "[", "]" на сбалансированность скобок.
Примеры входа и выхода функции:
Вход -> Выход
([]) -> true
(() -> false

[(]) -> false

)(-> false

[[[(([]))]][]()]] -> true

27. Задача 27

Реверс числа без использования строк

28. Задача 28

Реализация бинарного дерева

29. Задача 29

Проверка симметричности бинарного дерева

30. Задача 30

Дан массив с числами, одно число удалили и перемешали массив. Найти удаленное число.

31. Задача 31

Вычисление глубины дерева

32. Задача 32

Найти общее число в трёх коллекциях, не используя дополнительные структуры

33. Задача 33

Есть такой кусок кода, это правильный ответ на задачу,но условия - нет))))

```
class Solution {
    public int[] twoSum(int[] nums, int target) {
        HashMap<Integer, Integer> map = new HashMap<>();
        int[] ans = new int[2];
        for (int i = 0; i < nums.length; i++) {
            if (map.containsKey(target - nums[i])) {
                ans[0] = map.get(target - nums[i]);
                ans[1] = i;
                return ans;
            }
            map.put(nums[i], i);
        }
        return ans;
    }
}
```

34. Задача 34

```
/ Дан массив целых чисел, отсортированный по возрастанию.
// Вернуть массив, содержащий элементы исходного массива в квадрате,
// также отсортированный по возрастанию.

// [1, 4, 10] → [1, 16, 100]

// [-5, -3, 0, 1, 2, 4] → [0, 1, 4, 9, 16, 25]

//  [-5, -3, 0, 1, 2, 4]
```

```
//  [0, 1, 4, 9, 16, 25]

// [-1, 0, 1] → [0, 1, 1]

// 1. x^2 > 0
// 2. input arr is sorted

import java.io.*;
import java.util.*;


class MyCode {
  public static void main (String[] args) {
    // int[] in = {1, 4, 10};
    int[] in = {-9, -4};

    // [-9, -4, 10000]
    // [16, 81, 10000^2]

    System.out.println(Arrays.toString(square(in)));
  }

  public static int[] square(int[] inputArr){
    int[] output = new int[inputArr.length];
    int leftI = 0;
    int rightI = inputArr.length - 1;
    int second = inputArr.length - 1;

    while(leftI <= rightI){
      if(Math.abs(inputArr[leftI]) > Math.abs(inputArr[rightI])){
        output[second] = inputArr[leftI] * inputArr[leftI];
        leftI++;
      } else {
        output[second] = inputArr[rightI] * inputArr[rightI];
        rightI--;
      }
      second--;
    }
    return output;
  }

}
```

35. Задача 35

```
// // Необходимо написать функцию, которая получает строку с абсолютным UNIX путем
// // и возвращает укороченную версию, удалив все ненужное

// // Input: "/foo/../test/../test/../foo//bar/./baz"
// // Output: "/foo/bar/baz"
// // Пояснения

// // .. - возврат на директорию выше
// // . - означает текущую директорию, по сути просто мусор
// // /// - просто мусор
// // нельзя выйти выше корневой директории (/foo/../../)


// // /foo <- test <- test foo bar baz

// // Space O(n) // Time O(n)

// import java.io.*;
// import java.util.*;

// class MyCode {
//   public static void main (String[] args) {
//     Stack<String> stack = truePath("/foo/../test/../test/../foo//bar/./baz");
//     StringBuilder output = new StringBuilder();
//     for(String element : stack){
//       output.append("/");
//       output.append(element);
//     }
//     System.out.println(output.toString());
//   }

//   public static Stack<String> truePath(String input){
//     String [] stringArr = input.split("/");

//     // pop/push/empty
//     Stack<String> output = new Stack<>();

//     for(int i = 0; i < stringArr.length; i++){
//       if(stringArr[i].equals("..")){
//         if(output.isEmpty()){
//           throw new RuntimeException("Out of dir");
//         }
```

```
//        output.pop();
//        continue;
//      }
//      if(stringArr[i].equals("") || stringArr[i].equals(".")){
//        continue;
//      }
//      output.push(stringArr[i]);
//    }
//    return output;
//  }
// }
```