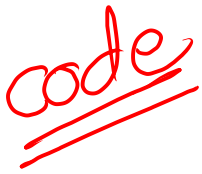
Classes and Objects 3_1




```
class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.n=n;
        this.arr=arr;
    }
}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution.
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();


        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }


        YoutubeVideos obj=new YoutubeVideos(song,artist,views,likes,releaseYear,n,arr);

        System.out.println(obj.song);
        System.out.println(obj.artist);
        System.out.println(obj.views);
        System.out.println(obj.likes);
        System.out.println(obj.releaseYear);
        System.out.println(obj.n);

        arr=obj.arr;

        for(int i=0;i<arr.size();i++){
            System.out.println(arr.get(i));
        }


    }
}
```

Constructor Overloading 3_1

Code

```
class youtubeVideo{
    String song;
    String artist;
    int views;
    int likes;
    int ryear;
    int n;
    ArrayList<String> arr=new ArrayList<>();

    youtubeVideo(String song,String artist,int views,int likes){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        System.out.println("The song has just released");
    }

    youtubeVideo(String song,String artist,int views,int likes,int ryear){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.ryear=ryear;
        System.out.println("The song was released in " + ryear);
    }

    public void getPrint(){
        System.out.println(song);
        System.out.println(artist);
        System.out.println(views);
        System.out.println(likes);
        System.out.println(ryear);
        System.out.println(n);
        for(int i=0;i<n;i++){
            System.out.println(arr.get(i));
        }

    }

}
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc=new Scanner(System.in);
        String song=sc.nextLine();
        String artist=sc.nextLine();
        int views=sc.nextInt();
        int likes=sc.nextInt();
        int ryear=sc.nextInt();
        int n=sc.nextInt();
        sc.nextLine();
        ArrayList<String> arr=new ArrayList<>();
        for(int i=0;i<n;i++){
            arr.add(sc.nextLine());
        }
        int k = sc.nextInt();
        youtubeVideo yvideo;
        if (k == 1)
            yvideo=new youtubeVideo(song,artist,views,likes);
        else
            yvideo=new youtubeVideo(song,artist,views,likes,ryear);
    }
}
```

Inheritance 3_1

code

```
class YoutubeVideo {
    String song;
    String artist;
    int views;
    int likes;
    int ryear;
    int n;
    ArrayList<String> arr=new ArrayList<>();
}

class ShortVideo extends YoutubeVideo {
    int time;
    int impression;

    ShortVideo(String song,String artist,int views,int likes,int ryear,int n,ArrayList<String> arr, int time, int
impression){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.ryear=ryear;
        this.n=n;
        this.arr=arr;
        this.time = time;
        this.impression = impression;
    }

    public void printProp() {
        System.out.println(song);
        System.out.println(artist);
        System.out.println(views);
        System.out.println(likes);
        System.out.println(ryear);
        System.out.println(n);
        for(int i=0;i<n;i++){
            System.out.println(arr.get(i));
        }
        System.out.println(time);
        System.out.println(impression);
    }
}
public class Solution {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String song=sc.nextLine();
        String artist=sc.nextLine();
        int views=sc.nextInt();
        int likes=sc.nextInt();
        int ryear=sc.nextInt();
        int n=sc.nextInt();
        sc.nextLine();
        ArrayList<String> arr=new ArrayList<>();
        for(int i=0;i<n;i++){
            arr.add(sc.nextLine());
        }
        int time = sc.nextInt();
        int impression = sc.nextInt();

        ShortVideo obj = new ShortVideo(song, artist, views, likes, ryear, n, arr, time, impression);
        obj.printProp();
    }
}
```

Public Private 3_1

Code

```
class youtubeVideo{
    String song;
    String artist;
    int views;
    int likes;
    int ryear;
    int n;
    ArrayList<String> arr=new ArrayList<>();
    private int impression;

    youtubeVideo(String song,String artist,int views,int likes,int ryear,int n,ArrayList<String> arr,int impression){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.ryear=ryear;
        this.n=n;
        this.arr=arr;
        this.impression=impression;
    }

    public void getImpression() {
        System.out.println(impression);
    }

    private int calcRank() {
        int sum = likes + impression;
        if (sum > 10000) {
            return 1;
        } else if (sum > 5000) {
            return 2;
        } else if (sum > 1000) {
            return 3;
        } else if (likes == 0) {
            return 4;
        }
        return 0;
    }

    public void printRank() {
        System.out.println(calcRank());
    }
}
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc=new Scanner(System.in);
        String song=sc.nextLine();
        String artist=sc.nextLine();
        int views=sc.nextInt();
        int likes=sc.nextInt();
        int ryear=sc.nextInt();
        int n=sc.nextInt();
        sc.nextLine();
        ArrayList<String> arr=new ArrayList<>();
        for(int i=0;i<n;i++){
            arr.add(sc.nextLine());
        }
        int impression = sc.nextInt();
        youtubeVideo yvideo=new youtubeVideo(song,artist,views,likes,ryear,n,arr,impression);
        yvideo.getImpression();
        yvideo.printRank();
    }
}
```

Method Overriding 3_1

Code

```
class youtubeVideo{
    String song;
    String artist;
    int views;
    int likes;
    int ryear;
    int n;
    ArrayList<String> arr=new ArrayList<>();

    public void getPrint(){
        System.out.println(song);
        System.out.println(artist);
        System.out.println(views);
        System.out.println(likes);
        System.out.println(ryear);
        System.out.println(n);
        for(int i=0;i<n;i++){
            System.out.println(arr.get(i));
        }
    }

    public void printTempRank() {
        System.out.println("Method of youtubeVideo class. Also can not find the rank right now.");
    }
}
class shortVideo extends youtubeVideo {
    shortVideo(String song,String artist,int views,int likes,int ryear,int n,ArrayList<String> arr) {
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.ryear=ryear;
        this.n=n;
        this.arr=arr;
    }
}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be na
        Scanner sc=new Scanner(System.in);
        String song=sc.nextLine();
        String artist=sc.nextLine();
        int views=sc.nextInt();
        int likes=sc.nextInt();
        int ryear=sc.nextInt();
        int n=sc.nextInt();
        sc.nextLine();
        ArrayList<String> arr=new ArrayList<>();
        for(int i=0;i<n;i++){
            arr.add(sc.nextLine());
        }
        shortVideo svideo=new shortVideo(song,artist,views,likes,ryear,n,arr);
        svideo.printTempRank();
    }
}
```

Check if array pair are divisible by K

arr = [7 , 9 , 4 , 14 , 5 , 3] K = 3

each element → rem freq

case 1

element 1 = 9 rem = 0

element 2 = 3 rem = 0

element1 = xK + 0

element2 = yK + 0

↓

Case 2 element1 = 14 , rem = 2

element2 = 4 , rem = 1

find rem1 = curr element

find rem2 = K − rem1

element1 = (4)(3) xK + 2 = 14

element2 = yK + 1 = 4

element1 + element2 = (1)(3) + 1 xK + yK + K = K(x+y+1)

arr = [7 , 9 , 4 , 14 , 5 , 3]

K = ~~6~~ 4

rem = 1 0 1 2 2 0

even

2 * 2 = 4

map

rem	freq
1	2
0	2
2	2

remainders freq

element 1 = 2K + rem1

rem1 = = K/2

rem1 + rem1 = K/2 + K/2 = K

Code

```java
public static boolean divisiblePair(int[] arr, int n, int k) {
    HashMap<Integer, Integer> map = new HashMap<>();
    for (int i = 0; i < n; i++) { // store freq of remainder
        int rem = arr[i] % k;
        map.put( rem, map.getOrDefault(rem, 0) + 1 );
    }

    for (int i = 0; i < n; i++) {
        int rem = arr[i] % k;
        if ( rem == 0 ) {
            int freq = map.get(rem);
            if ( freq % 2 == 1 ) {
                return false;
            }
        } else if ( 2 * rem == k ) {
            int freq = map.get(rem);
            if ( freq % 2 == 1 ) {
                return false;
            }
        } else {
            int freq1 = map.get(rem);
            int freq2 = map.getOrDefault( k - rem, 0 );
            if ( freq1 != freq2 ) {
                return false;
            }
        }
    }
    return true;
}
```

T.C = O(n)

S.C = O(n)

42. Trapping Rain Water

```java
class Solution {
    public int trap(int[] height) {
        int n = height.length;
        int[] leftMax = new int[n];
        int[] rightMax = new int[n];
        for (int i = 1; i < n; ++i)
            leftMax[i] = Math.max(height[i-1], leftMax[i-1]);

        for (int i = n-2; i >= 0; --i)
            rightMax[i] = Math.max(height[i+1], rightMax[i+1]);

        int ans = 0;
        for (int i = 0; i < n; ++i) {
            int waterLevel = Math.min(leftMax[i], rightMax[i]);
            if (waterLevel >= height[i]) ans += waterLevel - height[i];
        }
        return ans;
    }
}
```

Revision

- sorting lambda fn
- array
- 2 pointer
- Prefix array
- Array as hashmap
- 2d array
- String, substring
- Binary search (BSUB, BSLB)
- Arraylist
- stack
- Hashmap, Hashset
- Queue
- PQ

✓ sliding window

Lambda

- a - b // Ting order by value
- b - a // Ting " " "
- -1 // a first
- +1 // b first

Rotation right

3 step

1) $(n-k, n-1)$ reverse
2) $(0, n-k-1)$ reverse (linear)
3) $(0, n-1)$ reverse

gmp

$$k = k \% n$$

Kadane's algo

- max sum
- sum so far

ing

up and down

if (sum_so_far < 0)

start again

else

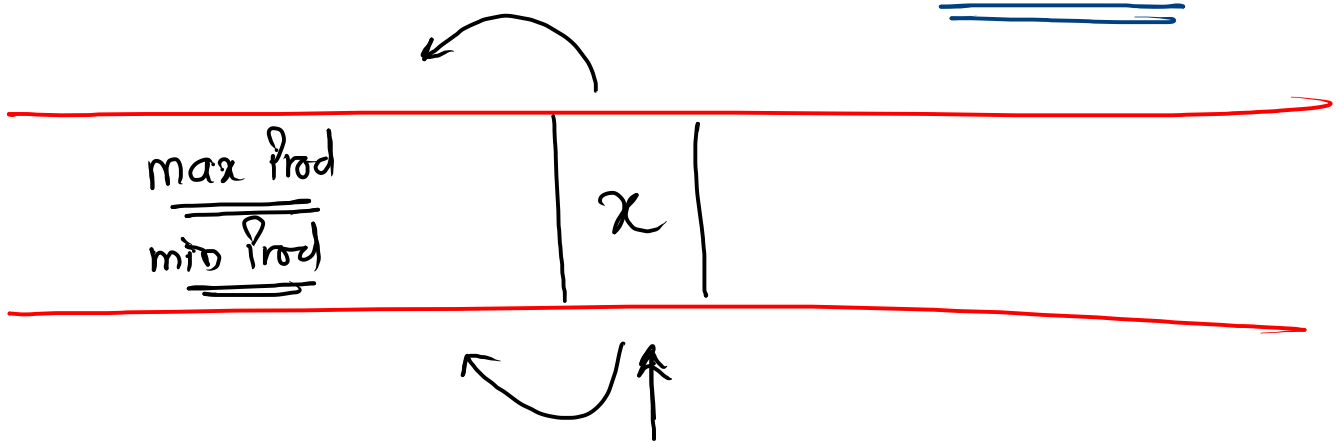
Cont. subarray

max sum

Quel

max Prod

wall = 0



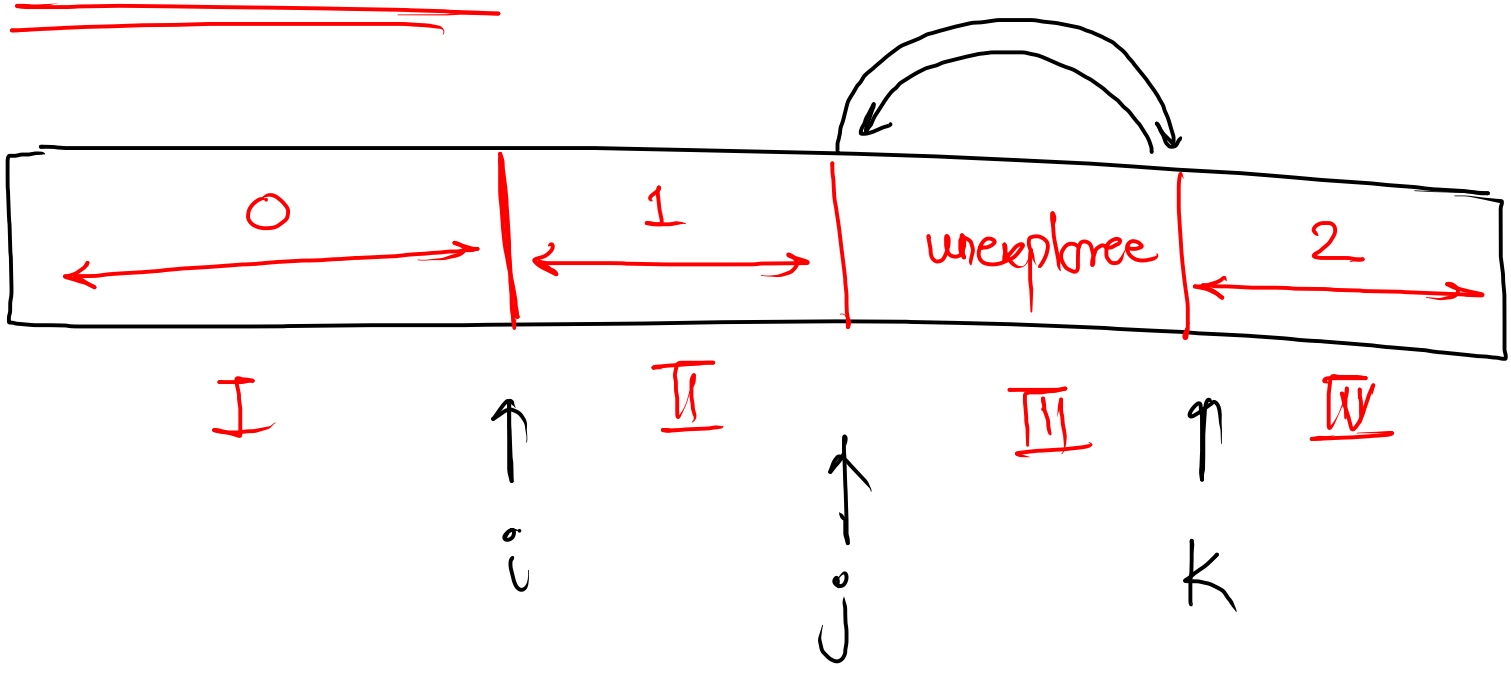
temp = maxProd

maxProd = max(curr, maxP * curr, minP * curr)

minProd = min (" , temp * curr ")

2 pointer

Sort 012



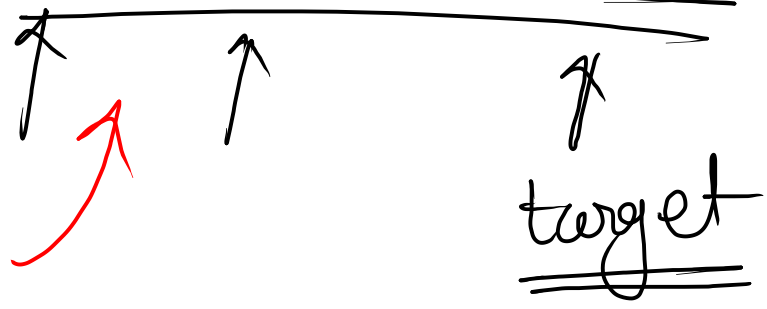
⇒ 3 sum

arr[i] + arr[j] + arr[k] = 0

precise

sort

arr[i] + arr[j] = -1 * arr[k]



target sum

(just run target sum code n time for each element.

⇒ Count bocat

limit = 5

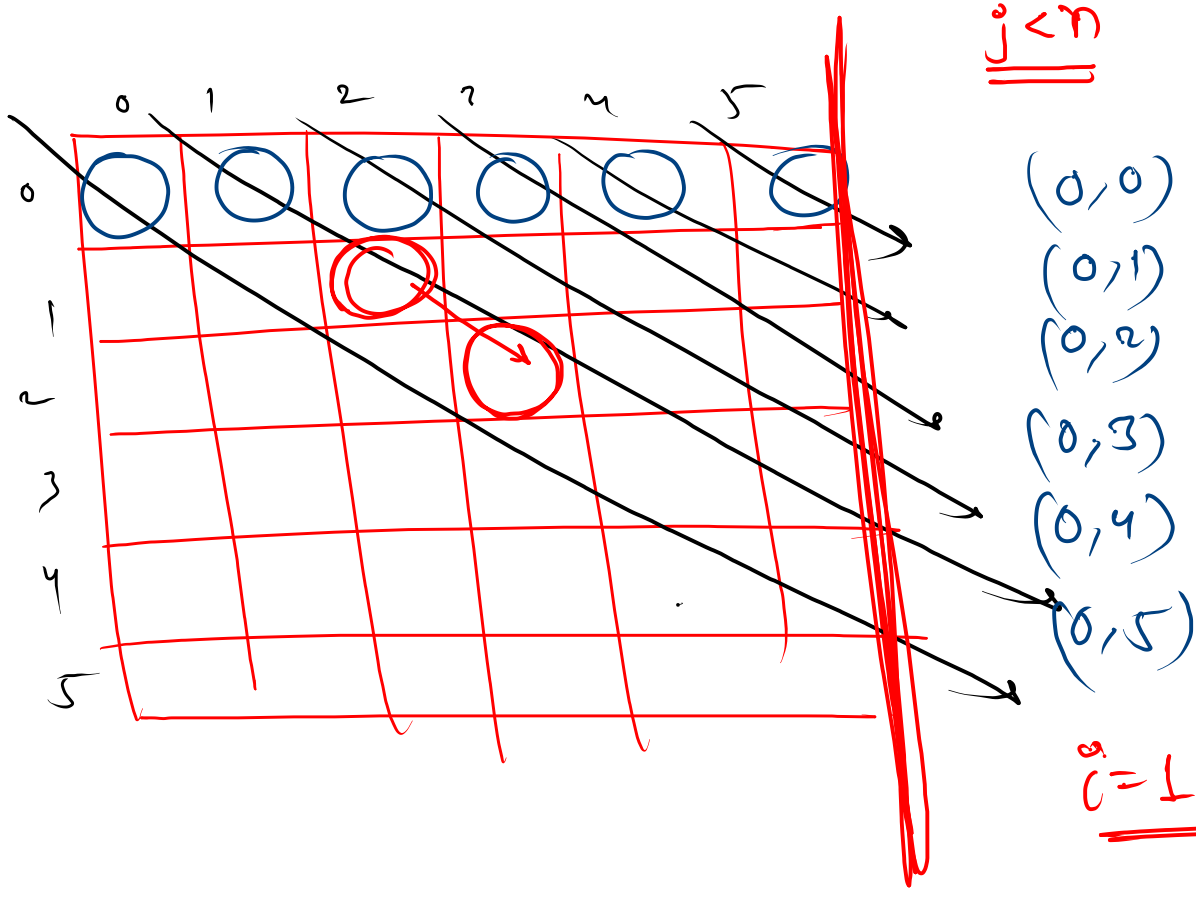
arr = [3, 5, 1, 2, 3]

T.C = $n \log(n)$

2d array

for(— , — , —)

j<n



(0,0)
(0,1)
(0,2)
(0,3)
(0,4)
(0,5)

i=1, j=0→n

Prefix , Suffix

- Prefix Sum
- Prefix Prod ✓
- Prefix min/max ✓

pre[i] = pre[i-1] + arr[i]

transpose

$(i > j)$

$(i < j)$

→ Convert 1D to 2D (m * n)

$$i = idx / n$$

$$j = idx \% n$$

→ 2D to 1D

$$idx = i * n + j$$