VPEX Solutions

By unrated

Problemsetters

Testers

Other

Do the problems bobliu

Problem 0 - Pizza

Time Limit: 2.0s **Memory Limit:** 64M

Sharing is caring, and **Darcy_Liu** loves to share. If he has x slices of pizza to give away and split evenly among ~~his~~our N friends, how many slices can he give to each, and how many will be left over?

Input

One line consisting of 2 integers x and N $(1 \leq x, N \leq 10^9)$.

Output

For each test case, output 1 line containing the number of slices each friend will get followed by the number of slices left over.

Solution

We are dividing x slices among N friends. Each friend gets x/N slices, rounded down. There are x mod N slices left over.

Time complexity: O(1)

Key concepts: simple math

Code (c++)

```cpp
#include <iostream>

using namespace std;

int main() {
  int a, b;
  scanf("%d %d", &a, &b);
  printf("%d %d\n", a/b, a%b);
}
```

VPEX '20 P1 - War on Two Fronts

Time Limit: 2.0s **Memory Limit:** 64M

Darcy has 10 classmates. 5 sit on the left side, while 5 sit on the right side. Each classmate has a certain amount of points. One day, Bruce invites Darcy to write on the blackboard. Facing attention from both the left and right side, he suddenly finds himself fighting a war on two fronts.

The situation may seem dire for Darcy, but luckily he has a secret ability: the blitzkrieg. In the blink of an eye, Darcy obliterates one classmate (using facts and logic), shocking the other 4 classmates on that side to surrender and give Darcy all their points. However, Darcy can only do this 1 time - as soon as he uses this ability, Bruce will banish him back to his seat for "being disruptive". How many points can Darcy win?

Input Specification

The first line contains 5 integers, with each integer x being the number of points of a classmate on the left side.

The second line contains 5 integers, with each integer x being the number of points of a classmate on the right side.

Constraints

$1 \leq x \leq 100$

Output Specification

Output the maximum amount of points Darcy can obtain.

Solution

The person Darcy attacks does not give any points, so Darcy should attack the person with the least points.

For each side, simply add up the points of every person and subtract the minimum. Then find the maximum of these 2 results.

Time complexity: O(1), as there are only 10 people in total

Key concepts: none

Code (c++)

```cpp
#include <bits/stdc++.h>

using namespace std;

#define inf 0x3f3f3f3f
int a, Min1 = inf, Min2 = inf, sum1, sum2;

int main() {
    for (int i = 0; i < 5; i++){
        cin >> a;
        Min1 = min(Min1, a);
        sum1 += a;
    }
    for (int i = 0; i < 5; i++){
        cin >> a;
        Min2 = min(Min2, a);
        sum2 += a;
    }
    cout << max(sum1 - Min1, sum2 - Min2) << "\n";
    return 0;
}
```

Code (Java)

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int rightsum=0;
        int rightmin=0x3f3f3f3f;
        int leftsum=0;
        int leftmin=0x3f3f3f3f;
        for(int i=0; i<5; i++) {
            int x = sc.nextInt();
            leftmin=Math.min(leftmin, x);
            leftsum+=x;
        }
        for(int i=0; i<5; i++) {
            int x = sc.nextInt();
            rightmin=Math.min(rightmin, x);
            rightsum+=x;
        }
        System.out.println(Math.max(rightsum-rightmin, leftsum-leftmin));
        sc.close();
    }
}
```

VPEX '20 P2 - Darcy Parties

Time Limit: 2.0s **Memory Limit:** 64M

Darcy is celebrating after winning the epic lOl victory royale. At his party, Darcy made some cake and split it equally among the guests, giving everyone an equal amount. Everybody there either enjoyed it or didn't enjoy it (you can't do both). What they didn't know was that Darcy was secretly hired by Eric to spy on potential enemies. During the party, Darcy began to notice something strange: those who enjoyed cake were beginning to BUY cake from those who didn't! Such a display shall not fool Darcy, for he has taken economics and knows this is an unacceptable instance of *capitalism* Help Darcy find out how many people have participated in this illegal activity.

Input specification

The first line contains N, the number of people at the party. The next line contains N integers, each integer x representing the amount of cake a person has.

Output specification

Output the number of people who have either sold or bought cake.

Constraints

$1 \leq N, x \leq 10$

Solution

Whenever a person sells cake to another person, the total amount of cake does not change. So you can find the amount of cake everyone starts off with by dividing the total amount by the number of people.

If someone sells cake, they will have less than they started with. If someone buys cake, they will have more. No one will buy and then sell because you can’t like and dislike cake at the same time. Count the number of people who do not have the amount they started off with to get your answer.

Time complexity: O(N)

Key concepts: simple math

Code (c++)

```cpp
#include <bits/stdc++.h>

using namespace std;

const int MM = 20;
int N, arr[MM], sum, a, cnt;

int main() {
    cin >> N;
    for (int i = 0; i < N; i++){
        cin >> arr[i];
        sum += arr[i];
    }
    a = sum / N;
    for (int i = 0; i < N; i++){
        if (arr[i] != a){
            cnt++;
        }
    }
    cout << cnt << "\n";
    return 0;
}
```

Code (Java)

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int a[] = new int[N];
        int sum=0;
        for(int i=0; i<N; i++) {
            a[i] = sc.nextInt();
            if(a[i] < 1 || a[i] > 10 || N < 1 || N > 10) System.out.println("error");
            sum += a[i];
        }
        sum /= N;
        int cnt=0;
        for(int i=0; i<N; i++)
            if(a[i] != sum) cnt++;
        System.out.println(cnt);
        sc.close();
    }
}
```

VPEX '20 P3 - Coding Club

Kevin can't teach programming this week because his school forgot to install an IDE. Instead, he has decided to entertain the members of PETCS by letting them watch the bouncing DVD screensaver meme. The screensaver consists of a DVD logo, which can be approximated as a rectangle with width A and height B, bouncing around a rectangular screen of width W and height H at a speed of 1 unit/second. When the logo touches a side of the screen, it bounces off such that the angle of incidence equals the angle of reflection. When the logo reaches a corner, its direction is simply reversed.

Kevin soon created the animation, but he once again realized that the computers at his school couldn't run code. Suddenly, he had an idea. What if, instead of running the program at school, Kevin could simply record a the animation until it started to repeat, and play it in a loop? If he chose the right section, no one would be able to tell the difference. What is the minimum length of the video he has to record?

Input specification

The first line contains integers W and H, the width and height of the screen.
The second Line contains integers A and B, the width and height of the logo.
The third line contains integers x_0 and y_0, representing the starting position of the logo (measured from the bottom left corner of the screen and logo).
The last line contains integers x and y, meaning the logo has the same initial direction as an arrow pointing from $(0, 0)$ to (x, y).

Output specification

Let T be the minimum length of video Kevin must record. Print the first 6 non-zero digits of T.

If the animation will never repeat, print `-1`.

Constraints

$1 \leq A < W \leq 1000$
$1 \leq B < H \leq 1000$
$1 \leq A + x_0 \leq W$
$1 \leq B + y_0 \leq H$
$-10^5 \leq x, y \leq 10^5$
$x \neq 0$ or $y \neq 0$

Subtask 1 [20%]

$1 \leq W, H, x, y \leq 15$

Subtask 1 Solution

The values are so small, it is enough to simply simulate the rectangle, changing its direction every time it bounces off a wall.

When the box has the same position and is travelling in the same direction as it started in, the animation will begin to loop.

Time complexity: about O(AB), depending on how you implement

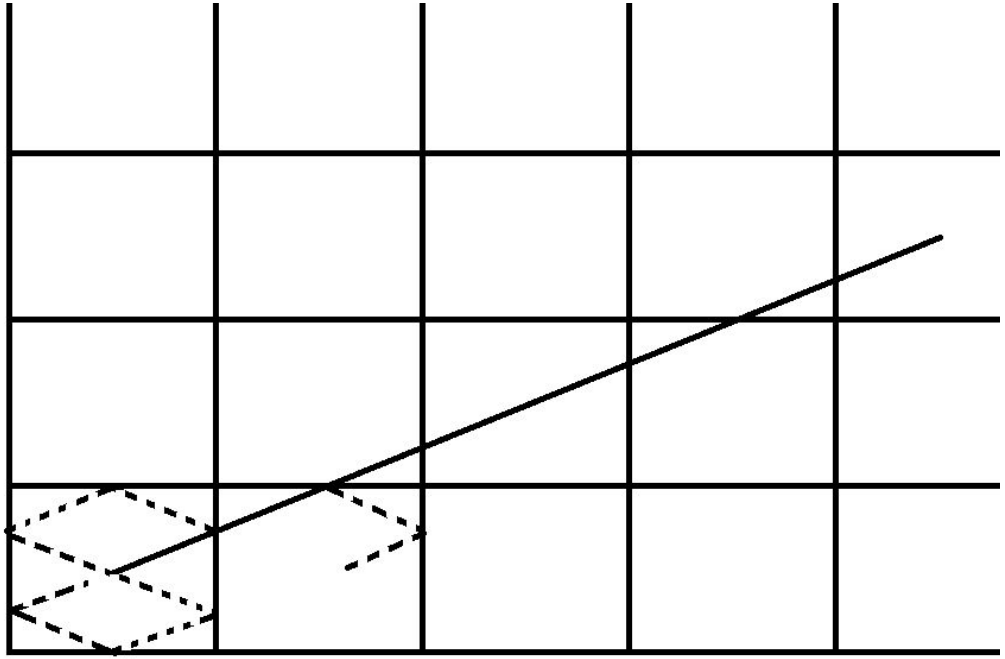
Key concepts: implementation

Full Solution

Instead of a AxB logo bouncing around a WxH screen, you can focus on the bottom left point bouncing around a (W-A)x(H-B) box. This can then be imagined as a point travelling in a line through an infinite grid of (W-A)x(H-B) cells.

When then point travels through an even number of cell vertically, it will have bounced an even number of times vertically. Same thing for the horizontal direction. When this has happened, the box will be travelling in its starting direction.

The box also needs to have travelled to its original position, so the vertical and horizontal displacement must be divisible by the cell height and width.



Full solution

If **a** is the horizontal displacement, and **b** is the vertical displacement, you have

$$a \bmod W = 0,\ a = mW$$

$$b \bmod H = 0,\ b = nH$$

From the slope, also

$$\frac{b}{a} = \frac{y}{x}$$

Then

$$\frac{nH}{mW} = \frac{y}{x}$$

$$yWm = xHn$$

$$= lcm(yW, xH)$$

because it's the first time this happens

So

$$a = \frac{lcm(yW,xH)}{y}$$

$$b = \frac{lcm(yW,xH)}{x}$$

The distance is

$$\sqrt{a^2 + b^2}$$

And travelling at one unit/second, it's also the time taken.

You can also see that the output will never be -1.

Remember the special cases where x=0 or y=0, where the point only travels in one direction.

Also remember to store your output as a double to avoid integer overflow.

Time complexity: O(1)

Key concepts needed: Simple math

Code (c++)

```cpp
inline ll gcd(ll a, ll b){return b==0?a:gcd(b,a%b);}
ll a,b,w,h,x,y,x0,Y0,z;
double t;

int main()
{
    cin>>w>>h>>a>>b>>x0>>Y0>>x>>y;
    w-=a; h-=b; w*=2; h*=2;
    if(x==0)
        z=w*1000000;
    else if(y==0)
        z=h*1000000;
    else
    {
        t=1.0*w*y*h*x/gcd(w*y,h*x);
        z=sqrt(t/x*t/x+t/y*t/y)*1000000;
    }
    string out=to_string(z);
    cout<<out.substr(0,6);
    return 0;
}
```

VPEX '20 P4 - Etopika

Time Limit: 2.0s **Memory Limit:** 128M

Bobliu the monkey is living on a tree. The tree can be modelled as a tree (a graph with N nodes and $N-1$ edges. Bob is currently on the ground, marked node 1. Every day, 2 new nodes (not always distinct) grow a banana, and Bob climbs from his current spot to the 2 bananas (in any order) and eats them. He then takes a nap where he is and sleeps until the next day. What is the least distance he must travel?

Input

The first line contains N, the number of nodes, and D, the number of days.

The next $N-1$ lines contain a, b, and c, marking a branch between a and b of length c.

The next D lines contain x and y, the location of the 2 bananas that day.

Output

Output the minimum total distance the monkey must travel.

Subtask 1

Calculate the distance between 2 nodes using BFS or DFS. Calculate the distance travelled for every order of visiting the nodes and find the minimum.

Time complexity: $N * 2^D$

Key concepts required: graph theory, brute force

Subtask 2

Precompute the distance between any two nodes by doing a DFS or BFS from each node.

If nodes a_1 and b_1 had bananas yesterday, and nodes a_2 and b_2 had bananas today, you have four options today:

a_1 to a_2 to b_2, a_1 to b_2 to a_2, b_1 to a_2 to b_2, b_1 to b_2 to a_2

This can be solved using dynamic programming.

If dp[d][0] = distance travelled up to day d and ending on node a_2, and dp[d][1] ending on node b_2,

$$dp[d][0] = min(dp[d-1][0] + dist[a_{d-1}][b_d], dp[d-1][1] + dist[b_d - 1][b_d]) + dist[b_d][a_d]$$

$$dp[d][1] = min(dp[d-1][0] + dist[a_{d-1}][a_d], dp[d-1][1] + dist[b_d - 1][a_d]) + dist[a_d][b_d]$$

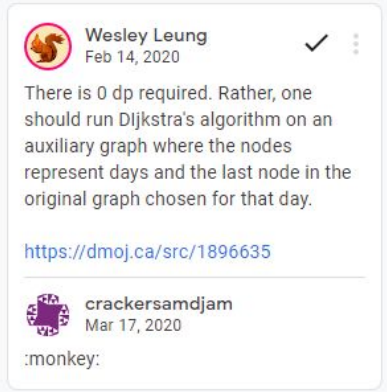
And the answer is $min(dp[D][0], dp[D][1])$

Because the monkey starts at node 1, $dp[1][0] = dist[0][a_1], dp[1][1] = dist[0][b_1]$

Time complexity: N^2+D

Key concepts: graph theory, dynamic programming

Full solution



This solution is similar to the solution to subtask 2. Instead of precomputing the distance between all pairs of nodes, we can find the distance between any two nodes by using their lowest common ancestor (LCA)

Time complexity: N+D

Key concepts: graph theory, dynamic programming, sparse table

Code (c++)

VPEX '20 P5 - Points Redistribution

Time Limit: 1.5s **Memory Limit:** 256M

After the revolution, all of Aaeria's points were taken away. Now he is working his way back up. On DMOJ there are N problems that take s minutes for him to implement and are worth v points. Each problem also has a topic. The 1st problem has topic 1, the 2nd problem has topic 2, and so on.

Unfortunately, due to Aaeria's poor programming abilities, he must rely on Bruce for help. This year there are Q classes. During each class, he is taught the topics from l to r and has t seconds to solve the problems. After the class, he looks at memes and forgets everything he learned.

After each class, the admins delete all of Aaeria's submissions, so they can be solved again for more points. However, due to a bug, he still keeps all the points he gained during the class.

Help Aaeria regain his former glory and calculate how many points he can obtain by the end of the year.

Subtask 1

For each query, solve the 0-1 knapsack problem with items between l and r

Key concepts: dynamic programming

Time complexity: max(r-l)*max(t)*Q

Subtask 2

Create a segment tree with each node storing the best value in range **l** to **r** for every time **t**. Merge the nodes in range during a query.

Key concepts: segment tree, dynamic programming, implementation

Time complexity: $Qt^2 * \log N$

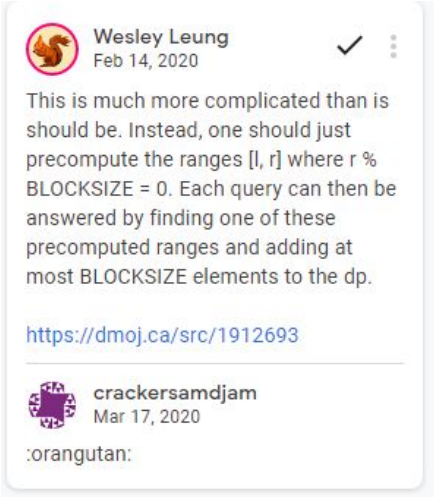
Full solution

It is possible to solve all queries from **l to r**, **l<=x<=r**, with 2 dp tables, **dp1[i][j]=** maximum value with weight **j in the range x+1 to i**, **dp2[i][j]=**maximum with weight **j** from **x-i to x**. This table takes **O(w(max(r)-min(l)))** to create. With a query from **l to r**, **ans=max(dp2[x-l][j]+dp[r][w-j])** for all **j<=w**. This query takes **O(w)** time.

Split the queries into 2 groups: ***r-l<=sqrt(n), r-l>sqrt(n)***. Sort both groups by ***l***. Solve the queries with the method above using initially x=0, until a query is reached where **l>x**, then set **x=r** and repeat until all queries are solved.

Key concepts: offline processing, square root decomposition, dynamic programming, implementation

Time complexity: $\mathbf{MN^{1.5}+MQ+Q \log Q}$



Code (c++)

Alternate solution

Begin with L=1, R=n, X=(L+R)/2. Solve all queries l to r where l<=X and r>X using the same method as the previous solution, a double ended dp array from X to L and X+1 to R. Then repeat the process with from L to X and from X+1 to R, and so on.

(Can be solved online with cdq table)

Key concepts: divide and conquer, dynamic programming, implementation

Time complexity: **MN*logN+Q(M+logN)**

https://github.com/Aaerialys/contest-programming/blob/master/dmoj/vpex1p5cdq.cpp

(if the link is broken, find it on my github at https://github.com/Aaerialys**)**

Rankings

	Name	School	Grade	Dmoj handle	Side (left/right)		
1	gay	gay	11	Resolute	LEFT IS BEST TTTTTT		
2		UTS	10	Dormi	right		
3		SJAM	10	Dan13llljws			
4		VPCI	11	CatGirl			
5		VPCI	10	GeeTransit			
6		VPCI	12	Justin			
7		VPCI	10	deadfish			
8		MGCI	12	Tzak			
9		VPCI	10	Avan			
10							
11		VPCI	10	kevinyang			
12		VPCI	10	YucenX			
13		VPCI	11	NewAccount			
14		VPCI	10	BryanJ			
16		VPCI	10	aarushjain33555			
17		mgci	11	dhrumilp15			

	Name	School	Grade	Dmoj handle	Side (left/right)		
18							
19		VPCI	10	Togohogo1			
20		VPCI	9	13lack13lood			
21							
22		VPCI	12	Vaaranan			
23							
24		VPCI	12	JimmyACookie			
25		VPCI	10	Stone_Yang			
26		VPCI	10	emilyyy333			
27		VPCI	9	jerryw			
28		VPCI	9	oskip123			
29		VPCI	10	jamsine			
30		VPCI	9	cyndie345678			
31		VPCI	11	franklai			
32		VPCI	10	madura30			
33		VPCI	10	ShengHong			
34		VPCI	9	asun18			

Left side wins

monkey

Bonus problem

VPEX '20 P5 - Topium

Time Limit: 2.0s **Memory Limit:** 64M

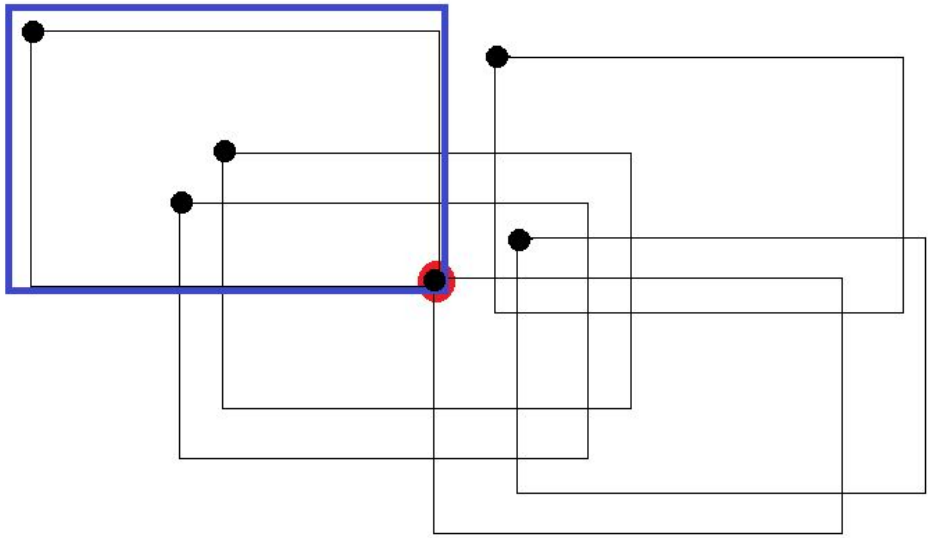
While exploring a distant planet, Amy discovered a new gem called Topium. Because of its many special properties, she wants to use it to create a window for her spaceship. To do this, she will need to cut out a R by C rectanglar plate.

Amy needs her ship to be able to withstand the harsh environments of space, and her biggest concern is the high temperature inside stars. After doing some experiments, she noticed that the gem consists of many small crystals arranged in a flat grid lattice with N rows and M columns. While most of these crystals are made of Topium, K of them, called **impurities**, are made of other materials of varying strength. The melting point of any gem fragment is equal to the sum of the strengths of all the impurities it contains. A pure fragment (containing no impurities) has a melting point of 0.

Amy can only cut the bonds between crystals, and not the crystals themselves. Help her determine the highest melting point of a fragment she can use for her window plate.

Solution

An impurity at ***(x,y)*** will be counted towards the melting point only if the top left corner of the plate is between **(x-R,y-C)** and **(x,y).**

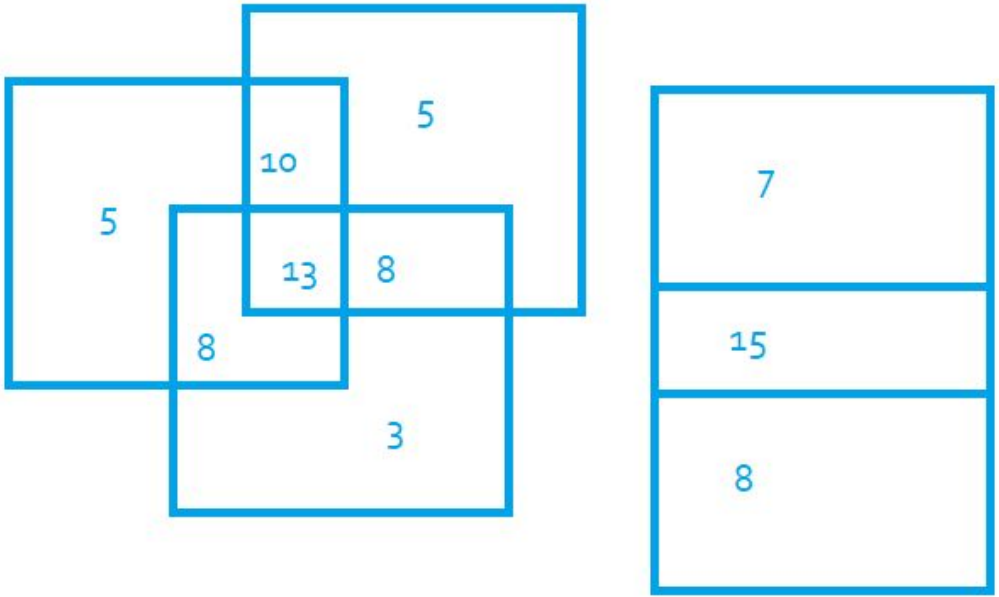


-

The plate will only contain the impurity (red) if its top left corner (black) is inside the blue box

Solution

Draw the bounding rectangle of each impurity. Each rectangle has a value equal to the impurity strength. Find highest possible total value of rectangles covering a point.



Solution

To solve this problem, store the left and right edges of each rectangle and sort them by x coordinate. Perform a line across the x axis. Store the value of each point on the y axis using a segment tree. When a left edge from **y1 to y2** of a rectangle of value **v**, add **v** to the segment tree from **y1 to y2.** When you reach a right edge, subtract v. Each time you add or subtract an edge, also find the maximum value in the segment tree. The maximum of these maximums will be your answer.

Coordinate compress the y values because they can be up to $\mathbf{10^9}$.

Tips:

- You cannot cut outside of the gem, all edges of your cut must be between **(1,1)** and **(n,m)**
- Some impurities have negative values so in some cases it may be the best option to look for a pure fragment
- Make sure you include all impurities inside your cut when adding up the values, even the negative ones

Solution

Key concepts: line sweep, segment tree, implementation

Time complexity: O(K*log M)

Code (c++)

redacted

redacted

redacted

redacted

redacted