Monotonous Queue

Monotonous queue is nothing but a neat trick which can be used is some specific problem. It helps reducing the run-time of many problems.

To describe Monotonous Queue I'll introduce you a simple problem.

Statement: n straight lines are given in the form $m_i x + c_i$. You'll be given m_i & c_i value for all lines from 1 to n. You'll be given Q queries each of which is a point x. You have to tell for each point x what is the best line i for which value of y (in $y = m_i \cdot x + c_i$) is minimum. Find that index i?

Constraint: $n <= 10^5$, $Q <= 10^5$

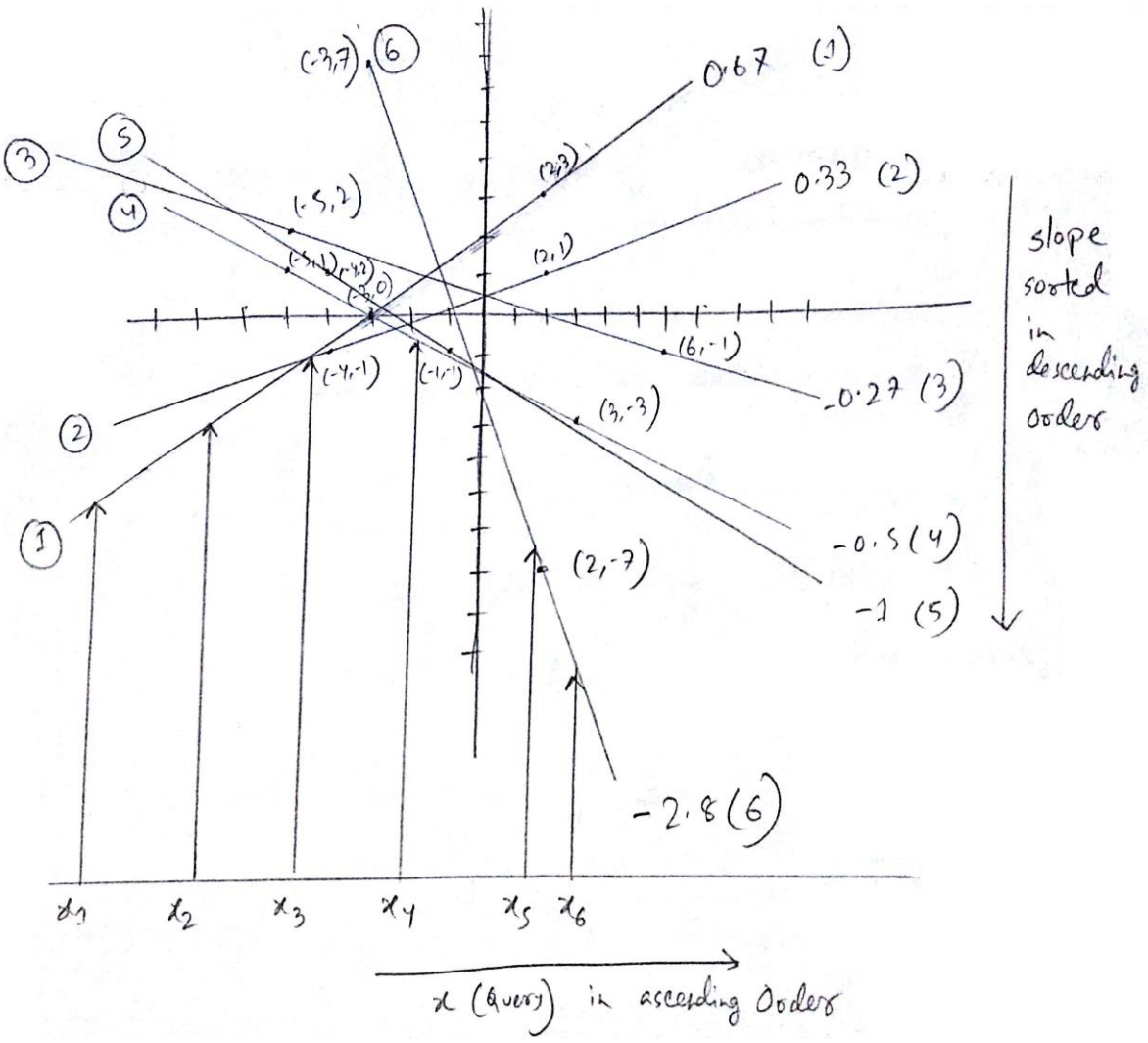
Solution 1: Naive approach.

for each query value x we check all the straight lines & get the lowest y value.

Complexity: $O(n)$ per query.

Solution 2: Convexhull trick.

step 1: as minimum y is asked sort all the slope of straight lines in descending order. We'll get this.


(-3,7)
6
0.67 (1)
3
5
(-5,2)
(2,3)
0.33 (2)
4
(-3,1)
(-2,0)
(2,1)
slope sorted in descending order
(6,-1)
-0.27 (3)
(-4,-1)
(-1,-1)
(3,-3)
2
1
-0.5 (4)
(2,-7)
-1 (5)
-2.8 (6)
x_1 x_2 x_3 x_4 x_5 x_6
x (query) in ascending order

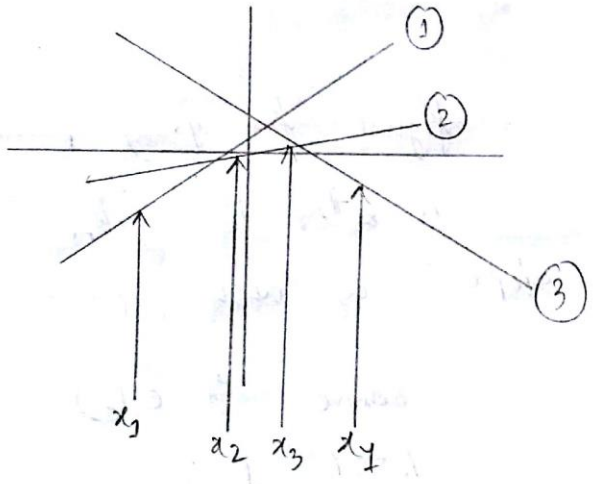
Now, suppose the query (x_i) values are sorted in ascending order.

Then for each x we can easily compute the value y.

Suppose, for x_{i-1} the required slope was $(M_t \cdot x_{i-1} + C_t)$

So, for x_i the required slope will be

between t & n ; [n is the total no. of slopes]

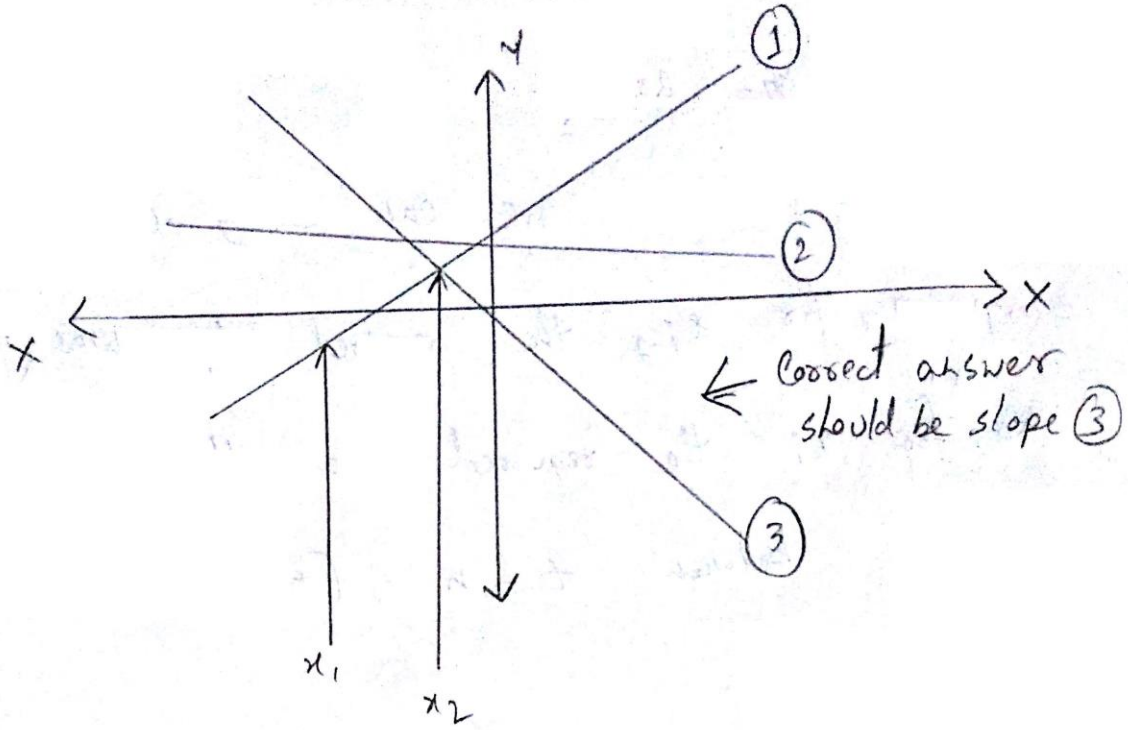


for query x_1 (1) slope gives minimum ; [compare both (1) & (2) (1) gives less]

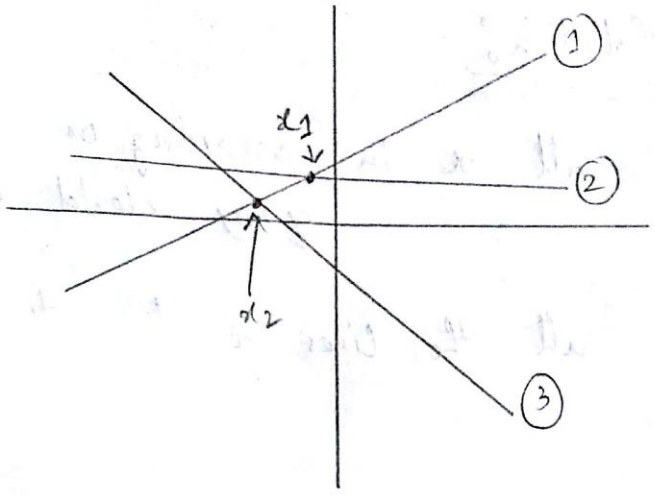
" query x_2 (2) " " " ; [compare with (1) & (2) (2) gives less]

" " x_4 (3) " " " ; [comp. both (2) & (3) (3) gives minimum]

But there is a problem.



in the above figure for query point x_2 we slope (1) is better than slope (2). Thus we get slope (1) as answer which is wrong. So, we need to remove slope (2) inorder get ourselves relieved of this situation. The solution is convexhull trick which is described below.



Intersecting Point of line ① & ⑪ $\Rightarrow m_1 x_1 + c_1 = m_2 x_1 + c_2$: [their y value is same]

$$\Rightarrow x_1 = \frac{c_2 - c_1}{m_1 - m_2}$$

Similarly Inter. Point of line ① & ⑭ $\Rightarrow$

$$x_2 = \frac{c_3 - c_1}{m_1 - m_3}$$

If $\boxed{x_2 >= x_1}$ then there is no x point where line ⑪ is an answer.
Thus line ⑪ should be removed.

Final Solution:

(i) Sort all x in ascending order i.e. query should be sorted.

(ii) Sort all the lines according to their slope in descending order.

(iii) Iterate through the lines to remove the unnecessary ones (convex hull trick)

(iv) Now, for each x we'll get a line (i) (query)

Compare with line (i+1) if that is better

if so, iterate through line (i+2), (i+3), ..., until you get a line which is best & then the next one.

Complexity: $n \log n + n \log n + n$

- $n \log n$: Sorting the slopes
- $n \log n$: Sorting the query
- n: Getting answer for each query.