Question: https://leetcode.com/problems/max-value-of-equation/

FIrst I tried the brute approach to find every possible value of yi+yj+|xi-xj|, and find the maximum but it gave TLE, so if we look carefully

Equation can be rearranged like this:  
=> yi+yj+|xi-xj|  
=> yi+yj+xj-xi (since xi < xj)  
=> (xj+yj)+(yi-xi)

Reason to use max heap or priority queue:  
if 2nd part(yi-xi) of the equation increases then value of total equation will also increase that's why used priority queue with (yi-xi) as key. xi is stored as value because that's required in next iteration to verify this condition : xj-xi > k

Code:  
class Solution {

public int findMaxValueOfEquation(int[][] points, int k) {

int res = Integer.MIN\_VALUE;

PriorityQueue<Pair<Integer, Integer>> pq = new PriorityQueue<>((a,b)->(b.getKey()-a.getKey()));

for(int[] point:points){

while((!pq.isEmpty())&&point[0]-pq.peek().getValue()>k) pq.poll();

if(!pq.isEmpty()){

res=Math.max(res, pq.peek().getKey()+point[0]+point[1]);

}

pq.offer(new Pair<>(point[1]-point[0], point[0]));

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>