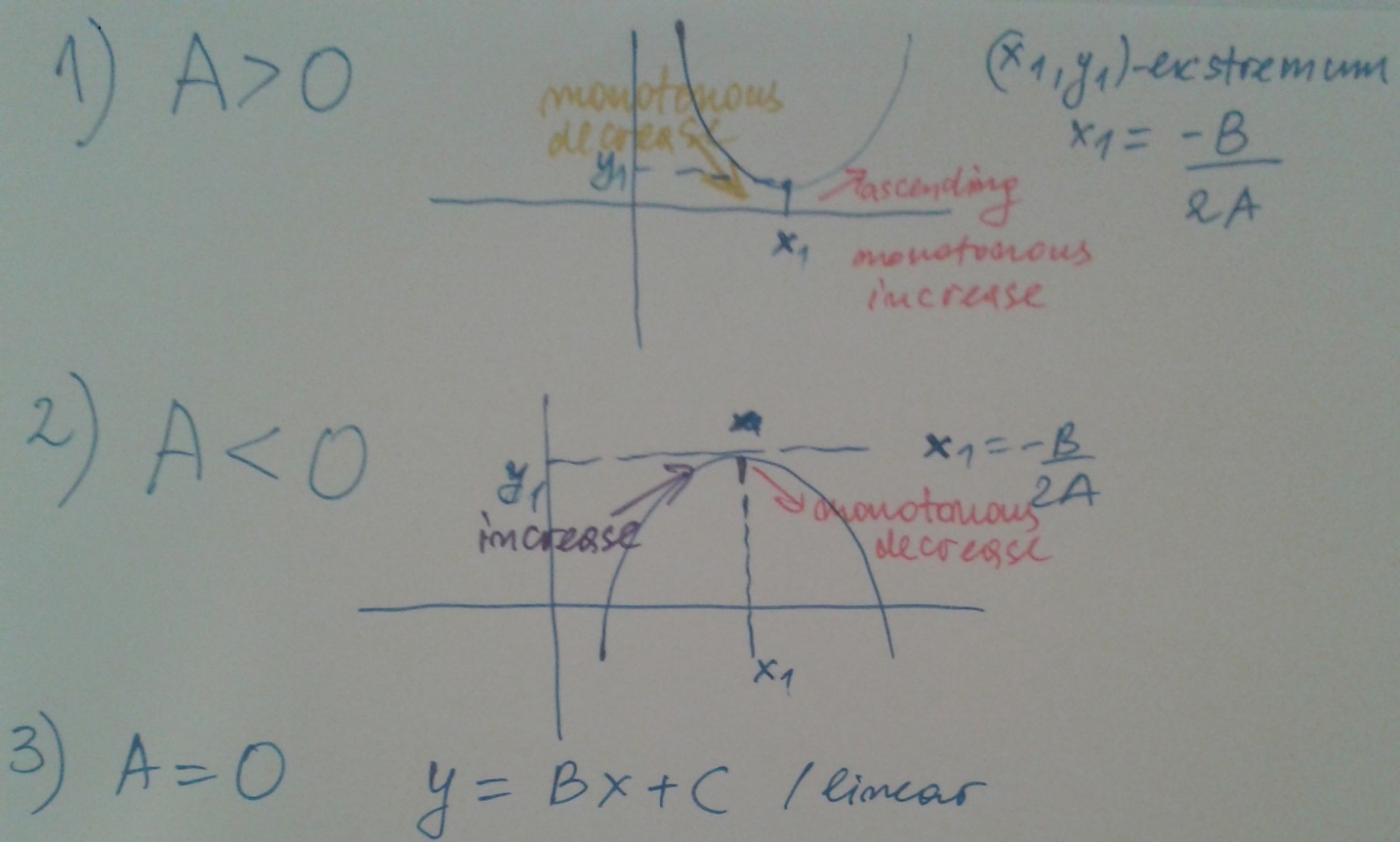
<https://massivealgorithms.blogspot.com/2016/05/sort-transformed-array.html>

<https://discuss.leetcode.com/topic/128/sort-transformed-array>  
Given a sorted array, and integer values A, B and C, return a new array after applying the equation x' = Ax^2+Bx+C to each element in the array.  
The returned array should be sorted.  
Expected time complexity : O(n)

I draw a picture that describe a little bit math, then will propose an algorithm for this problem. I must excuse myself that I am terrible painter, so don't mock at me.[](https://discuss.leetcode.com/uploads/files/1464360732032-img.jpg)  
As you see from the picture we have three different cases of the square function depending on the value of A.  
The extreme point of the parabola in case 1(minimum) and 2 (maximum) is x\* = -B /2A and y = Ax\*^+Bx\*+C

1. A > 0  
   You notice from the draft that when we have monotonous decreasing function for all x ,where x < x\* and vice verse monotonous decreasing function for all x > x\*, We can use this fact and split input array in two subarrays a and b :  
   a = [x0,x1,x2,x3....xk.<=x\*]  
   b = [x\* < xk+1, ....xn]  
   We calculate f(a) = [f(x0,f(x1)....f(xk)] and f[b] = [f(xk+1,f(xk+1)....f(xn)] , both arrays are sorted but in different order, one ascending, the other descending. We can revert the order of the descendin array and to merge them.
2. A < 0, the same as case 1, see the picture
3. A = 0, function is monotonous increasing , linear function, no need to merge

Based on A being -ve or positive, we can go ahead in two ways.

1. If A is +ve, it's upright parabola. Keep two pointers at both ends. Get x` = Ax^2 + Bx + C for both values, then put the maximum of the values to the end of the new array. Decrement or increment the pointers based on which element was selected.
2. If A is -ve, it's inverted parabola. This time, whichever element is lesser, put it to the start of the array