CS21003 ALGORITHMS-1
WorkSheet 7 Solutions
Date: 7 Nov 2020

1 Magic Distance

Lets get rid of the abs function.

$$MagicDistance((X_1, Y_1), (X_2, Y_2)) = max((X_1 - X_2) - (Y_1 - Y_2), (Y_1 - Y_2) - (X_1 - X_2))$$

Now rearrange the terms

$$MagicDistance((X_1, Y_1), (X_2, Y_2)) = max((X_1 - Y_1) - (X_2 - Y_2), (X_2 - Y_2) - (X_1 - Y_1))$$

Now, if X_1 and Y_1 are fixed, $X_1 - Y_1$ is fixed. So, the first term in the max function i.e. $(X_1 - Y_1) - (X_2 - Y_2)$ can be maximized if $X_2 - Y_2$ is minimized. Similarly, the second term is maximized if $Y_2 - X_2$ is minimized i.e. $X_2 - Y_2$ is maximized. So, to solve this, we maintain a min heap and max heap for $X_i - Y_i$. The answer for each Type 2 query (X, Y) can be found as $max((X - Y) - minheap.top(), maxheap.top() - (X - Y)$

Add - This just takes the point as input and adds the point to the minheap and maxheap using $X - Y$ as key and Y as value.

Remove - For this, we use two maps DeleteMapMin and DeleteMapMax to denote the status of points in heap. If we get a request to remove a point (X, Y), we don't directly delete the point from the heaps, instead, we will add the point to the DeleteMapMin and DeleteMapMax. If eventually the point (X, Y) comes to the top of maxheap or minheap, it will be deleted from the heap and then removed from the DeleteMapMax or DeleteMapMin respectively.

FindFarthest - It deletes the top element from the minheap and maxheap until the top element is not present in the DeleteMapMin and DeleteMapMax respectively. Then it returns the farthest point according to $max((X - Y) - minheap.top(), maxheap.top() - (X - Y)$