

## Minimum steps needed to cover a sequence of points on an infinite grid.

Given an infinite grid, initial cell position (x, y) and a sequence of other cell position which needs to be covered in the given order. The task is to find the minimum number of steps needed to travel to all those cells.

**Note:** Movement can be done in any of the eight possible directions from a given cell i.e from cell (x, y) you can move to any of the following eight positions: (x-1, y+1), (x-1, y), (x-1, y-1), (x, y-1), (x+1, y-1), (x+1, y), (x+1, y+1), (x, y+1) is possible.

**Approach:** Since all the given points are to be covered in the specified order. Find the minimum number of steps required to reach from a starting point to next point, then the sum of all such minimum steps for covering all the points would be the answer. One way to reach from a point (x1, y1) to (x2, y2) is to move  $\text{abs}(x2-x1)$  steps in the horizontal direction and  $\text{abs}(y2-y1)$  steps in the vertical direction, but this is not the shortest path to reach (x2, y2). The best way would be to cover the maximum possible distance in a diagonal direction and remaining in horizontal or vertical direction.

If we look closely this just reduces to the maximum of  **$\text{abs}(x2-x1)$**  and  **$\text{abs}(y2-y1)$** . Traverse for all points and summation of all diagonal distance will be the answer.

```
public class Solution {

    public static int getMinDistance(int x1,int y1,int x2,int y2){

        int dx = Math.abs(x2-x1);

        int dy = Math.abs(y2-y1);

        return Math.max(dx,dy);

    }

    public int coverPoints(ArrayList<Integer> A, ArrayList<Integer> B) {

        int length = A.size();

        int i;

        int min_distance = 0;
```

```
for(i=0;i<length-1;i++)  
{  
    int x1,x2,y1,y2;  
    x1 = A.get(i);  
    y1 = B.get(i);  
    x2 = A.get(i+1);  
    y2 = B.get(i+1);  
    int min_distance_between_points= getMinDistance(x1,y1,x2,y2);  
    min_distance += min_distance_between_points;  
  
}  
  
return min_distance;  
}  
}
```