

## Contact Tracing!

### Computer Science I Program #3: Sorting and Searching

**Please check Webcourses for the Due Date**

**Read all the pages before starting to write your code**

You are spending most of your time at home in this pandemic. It is of most importance for people to be aware of where other people, who are infected with COVID-19 are, and who they've been near. Keeping track of this information is known as "contact tracing." You've heard that there might be some very high paying jobs if you can show your contract tracing skills to the government, so you've decided to write a little program to highlight those skills.

Your area can be modeled on the Cartesian plane. You are located at the point  $(x, y)$ . In addition, you have the Cartesian coordinates of all people currently infected with COVID-19. What you would like to do is write a program that sorts these locations based on their distance from you, followed by handling queries. The queries are of the form of a point you are thinking of visiting. Your program should identify if someone who is infected is at that location, and if so, what their rank is on the sorted list of infected people. If no one is infected at that location, you should correctly identify this.

**Note: There are many important implementation restrictions for this assignment, so to make sure everyone reads these, the section on implementation restrictions will be next, changing the order of the sections as compared to other assignments.**

#### Implementation Restrictions

1. You must use a specified combination of Merge Sort and Insertion Sort to sort the point data. Specifically, for each input case, a threshold value,  $t$ , will be given. If the subsection of the array to sort has  $t$  or fewer values to sort, Insertion Sort should be used. Otherwise, Merge Sort should be used. Further details about the comparison used for the sorting are below.
2. You must store your coordinates in a struct that contains two integer fields.
3. You must write a function `compareTo` which takes in two pointers, `ptrPt1` and `ptrPt2`, to coordinate structs and returns a negative integer if the point pointed to by `ptrPt1` is closer to you than the point pointed to by `ptrPt2`, 0 if the two locations pointed to by both are identical locations, and a positive integer if the point pointed to by `ptrPt1` is farther from you than the point pointed to by `ptrPt2`. Exceptions to this will be when the two pointers are pointing to points that are the same distance from you, but are distinct points. In these cases, if `ptrPt1`'s x coordinate is lower than `ptrPt2`'s x coordinate, a negative integer must be returned. Alternatively, if `ptrPt1`'s x coordinate is greater than `ptrPt2`'s x coordinate a positive integer must be returned. Finally, if the x coordinate of both points is the same, if `ptrPt1`'s y coordinate is lower than `ptrPt2`'s y coordinate, a negative integer must be returned. If `ptrPt1`'s y coordinate is greater than `ptrPt2`'s y coordinate, a positive integer must be returned.

4. Since your location must be used for sorting, please make the variable that stores your x and y coordinates global. Your program should have no other global variables.
5. A Binary Search function must be used when answering queries.
6. Your sort function should take in the array to be sorted, the length of the array as well as the threshold value,  $t$ , previously mentioned. This function should NOT be recursive. It should be a wrapper function.
7. The recursive sort function (you can call this mergeSort) should take in the array, a starting index into the array, an ending index into the array and the threshold value  $t$ . In this function, either recursive calls should be made OR a call to an insertion sort function should be made.

### **The Problem**

Given your location, and the location of each person who has COVID-19, sort the list by distance from you from shortest to longest, breaking ties by x-coordinate (lower comes first), and then breaking those ties by y coordinate (lower comes first).

After sorting, answer several queries about points in the coordinate plane. Specifically, determine if a query point contains someone who is infected or not. If so, determine that person's ranking on the sorted list in distance from you.

### **The Input (to be read from standard input) - Your Program Will Be Tested on Multiple Files**

The first line of the input contains 5 integers separated by spaces. The first two of these values are  $x$  and  $y$  ( $|x|, |y| \leq 10000$ ), representing your location. The third integer is  $n$  ( $2 \leq n \leq 10^6$ ), representing the number of infected people. The fourth integer is  $s$  ( $1 \leq s \leq 2 \times 10^5$ ), representing the number of points to search for. The last integer,  $t$  ( $1 \leq t \leq 30$ ), represents the threshold to be used for determining whether you run Merge Sort or Insertion Sort.

The next  $n$  lines of the input contain  $x$  and  $y$  coordinate values, respectively, separated by spaces, representing the locations of infected people. Each of these values will be integers and the points will be distinct (and also different from your location) and the absolute value of  $x$  and  $y$  for all of these coordinates will not exceed 10,000.

Then the next  $s$  lines of the file contains  $x$  and  $y$  coordinate values for searching. Both values on each line will be integers with an absolute value less than or equal to 10,000.

### **The Output (to be printed to standard out)**

The first  $n$  lines of output should contain the coordinates of the people infected, sorted as previously mentioned. These lines should have the x-coordinate, followed by a space, followed by the y-coordinate.

The last  $s$  lines of output will contain the answers to each of the  $s$  queries in the input. The answer for a single query will be on a line by itself. If the point queried contains an infected person, output a line with the following format:

```
x y found at rank R
```

where  $(x, y)$  is the query point, and  $R$  is the one-based rank of that infected person in the sorted list. (Thus,  $R$  will be 1 more than the array index in which  $(x, y)$  is located, after sorting.)

If the point queried does NOT contain an infected person, output a line with the following format:

```
x y not found
```

### **Sample Input (Note: Query points in blue for clarity.)**

```
0 0 14 5 5
3 1
-6 -2
-4 3
4 -4
2 4
-1 3
2 2
0 -5
-4 -2
-6 6
4 4
-2 4
0 5
-4 6
2 -1
3 1
0 -5
0 5
-6 7
```

### **Sample Output**

```
2 2
-1 3
3 1
-4 -2
-2 4
2 4
-4 3
0 -5
0 5
4 -4
4 4
-6 -2
-4 6
-6 6
2 -1 not found
3 1 found at rank 3
0 -5 found at rank 8
0 5 found at rank 9
-6 7 not found
```

### **Deliverables**

You must submit a two files over WebCourses:

- 1) A source file, *contact.c*. Please use stdin, stdout. There will be an automatic 10% deduction if you don't do so.
- 2) Your Process Document, which should be named LASTNAME\_PROCESS.doc (or .docx or .pdf or .txt)