# CSE 204 January 2021: Data Structures and Algorithms I Sessional

## Divide and Conquer

Sherlock and Watson are longtime friends. Recently, Sherlock has faked his death to save Watson. Watson does not know about this. He is looking for a house in Manhattan and he comes to Mycroft for suggestions. But Watson does not know, Mycroft is helping Sherlock. Sherlock wants to stay close to Watson but not too close to avoid getting found out. So, Sherlock has asked Mycroft to find two houses for Sherlock and Watson.

There are $n$ houses in Manhattan. Each house $h_i$ has a two-dimensional position $(x_i, y_i)$. The distance between two houses $(h_i, h_j)$ is the Euclidean distance of $(x_i, y_i)$ and $(x_j, y_j)$.

Mycroft has calculated distances between every pair of houses. He has sorted the pairs by ascending order of distances and shows the result to Sherlock. Sherlock does not want the nearest pair. He picks the second nearest pair. However, Sherlock is surprised why Mycroft took an $O(n^2)$ approach. Sherlock can do it in $O(n \log^2 n)$. He asks you if you can do it in $O(n \log n)$.

**Input:**

The first line contains the integer $n$ $(3 \le n \le 10^5)$

The next $n$ lines have two integers, showing $x_i$ and $y_i$ of house $h_i$ $(-10^4 < x_i, y_i < 10^4\ ,\ 0 \le i < n)$

**Output:**

The first line contains two integers, $j$ and $k$ $(j < k)$, showing the indices of the second nearest houses of Manhattan.

The second line contains a number rounded to 4 decimal points showing the distance between these two houses ($h_j$ and $h_k$). This distance is guaranteed to be higher than the distance between the nearest pair.

| **Input** | **Output** |
|---|---|
| 3 | 1 2 |
| 0 0 | 2.0000 |
| 1 0 | |
| 3 0 | |
| 10 | 1 9 |
| -2289 9295 | 2020.4381 |
| 3137 3748 | |
| -5033 5679 | |
| 7729 -432 | |
| 9262 16 | |
| 8341 -5462 | |
| -4163 3577 | |
| 1710 -3328 | |
| -8676 -3161 | |
| 5106 3295 | |
| 5 | 0 1 |
| -9765 2172 | 7124.2921 |
| -4808 7289 | |
| 955 -2187 | |
| 9279 -9856 | |
| 6332 -2249 | |

**Algorithm Constraints:**

You have to take a *divide and conquer* approach.

The complexity of your algorithm must be $\boldsymbol{O(n \log n)}$

You have to write a report showing the complexity analysis of your code, especially the merge step of your divide and conquer algorithm.

**Special Instructions:**

1. Take input from a file.
2. Write *readable, re-usable, well-structured, quality code*. This includes but is not limited to writing appropriate functions for implementation of the required algorithms, meaningful naming of the variables, suitable comments where required, proper indentation, etc.
3. Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet, etc.).
4. Implement the algorithms with your style of coding. Any form of plagiarism (irrespective of source or destination), will result in getting **-100% marks**. You have to protect your code.
5. Also, be informed that for the repeated offense of plagiarism, the departmental policies suggest stricter measures.

**Submission Guideline:**

1. Create a directory with your 7 digit student id as its name
2. Put the source files and report into the directory created in 1
3. Zip the directory
4. Upload the zip to the Moodle

For example, if your student id is 1805123, create a directory named 1805123. Put your source files (.c, .cpp, .java, .py, .h, .hpp etc) and report (.docx, .pdf) into 1805123. Zip 1805123 into 1805123.zip and upload the 1805123.zip to the Moodle.

Failure to follow the above-mentioned submission guideline will result in up to 10% penalty.

**Submission Deadline:**

11:55 PM, 18th June 2021