

SSN College of Engineering

Department of Computer Science and Engineering

CS1403 — Design and Analysis of Algorithms

2019 – 2020

Session — 04

January 18, 2020

-
- This homework is due by 4pm on January 18, 2020
 - Grace period may be given up to midnight of January 19, 2020
 - You can upload only one ZIP file
 - The naming convention is “<Your first name (first letter capital and all the other letters small)>-CS1403-S04.zip”
 - The questions marked as “OPTIONAL” are, as the name implies, optional! Complete your core assignment first and attempt the optional problem only if you have sufficient time.
-

1. In a machine learning project, there is a need to maintain a list of example instances encoded as two dimensional vectors. For the sake of simplicity, it may be assumed that a list of two dimensional points are maintained.

When a new instance is given, i.e. when a new point $P_{new} = (x_{new}, y_{new})$ is given, it should be compared with all the points in the list by considering the distances between P_{new} and every P_i in the list. What is required is the list of points sorted according to their distances from the new reference point given.

For the purpose of this assignment, we will assume that the points are available through a Queue interface (a streaming source). Implementations of Queue and Point classes are provided for your convenience. You should not modify these classes. You can add a section in your report, if you find bugs in them or you have compelling reasons to change them.

- (a) Design and implement (in Python) an algorithm, with average time complexity not exceeding $O(n \log n)$, to produce the result queue from a given queue of points and a new reference point. You may assume that the given queue of points is a streaming source and need not be preserved. Note that the result queue contains all and only the points in the original queue, but sorted according to their distances from the reference point.
- (b) Analyse your algorithm and report additional space used (if any) in asymptotic notation.
- (c) (OPTIONAL) After this implementation, your client tells you that they only need the median and the entire sorted list is not required. Can you quickly write a function to return the median, using your current implementation?

- (d) (OPTIONAL) Now the client has what they want, but you wish to optimize the implementation! (you are a good software engineer, aren't you!?). Is it possible to find the median with time complexity better than $O(n \log n)$?
2. In a related project, a similar list of two dimensional points are maintained. It may be assumed that the list is available in an array like structure in the main memory. Each point refers to the location of an ATM of a particular bank. The bank is in a cost cutting mode and wishes to close down some of their ATMs. To start with they wish to identify a pair of ATMs that are very close to each other and remove one of them.
- (a) Design and implement (in Python) an algorithm to find the closest pair of ATMs from a given list of co-ordinates of ATMs. For simplicity, you may use brute-force approach to perform an exhaustive search.
 - (b) Analyse your algorithm and report the time complexity in asymptotic notation.
 - (c) (OPTIONAL) Perform empirical analysis to verify the time complexity.
 - (d) (OPTIONAL) Think if it is possible to use divide-and-conquer strategy to improve the time complexity. If possible, provide an outline of such an algorithm.