Individual Assignment 1 - Drone Collision Prevention

AI1225 - Algorithms and Data Structures

1 Introduction

The objective of this assignment is to design and implement a Python application that processes drone location data and determines pairs of drones that are close to each other based on their spatial coordinates. Variations of the problem are also included in the task description in subsequent sections. For simplicity, we assume that drone locations are provided as a *snapshot*, meaning, that drones do not move with time.

2 Data

To simulate a real-world scenario, first you need to generate drone locations. Keep in mind that you will work with both 2-dimensional and 3-dimensional data. Also note that the number of drones will vary in each experiment. The datasets you need to generate contain a unique drone ID, followed by a location either in 2D or 3D. You need to decide about the width of the dimensions, e.g., 1000x1000, 10000x10000, etc. Note that drone locations should be represented as pairs or triples of floats. You may store the datasets in files (e.g. csv) in order to reuse them in subsequent runs of your application. Note also that you need to test your application for different number of drones: 1,000 - 10,000 - 100,000 - 1,000,000 - 10,000,000). (*Note: in case the baseline approach runs for a large amount of time and never completes, you may omit running the baseline over large datasets and run only the optimized algorithms.*)

3 Required Tasks

Task 1 [40 marks]

The data for this task are 2D. **You are asked to detect the pair of drones that are closest to each other than any other pair**. Find a way to resolve ties if exist. You need to provide one baseline solution to the problem, report its complexity, and demonstrate its performance for different numbers of drones mentioned previously. Next, you need to provide a better algorithm to solve the problem, provide a proof of its correctness, and demonstrate its performance by comparisons with the baseline. Include a table with the execution times and a plot of the results showing the trend. Include a worst-case complexity analysis of your solution. Clearly state your observations in the report.

Task 2 [30 marks]

In this task, you need to repeat the process of Task 1, but now drone locations are represented in the 3D space. **You are asked to detect the pair of drones that are closest to each other than any other pair**. Find a way to resolve ties if exist. Again, you need to report the baseline and compare it with a more sophisticated algorithm that is better than the baseline, using both execution times and complexity arguments. Do not forget the proof of its correctness. Comparisons should include execution times for different numbers of drones. Include a table with the execution times and a plot of the results showing the trend. Clearly state your observations in the report.

Task 3 [15 marks]

In this task, you are called to extend the solutions provided in Task 2 (3D case), in order to report not only the pair with the minimum distance but the top-k pairs, for different values of k. Again, you need to test your solution for different numbers of drones as previously. Report exact running times in tables and provide plots to show the trend. Compare your solution to the baseline algorithm both using execution times and worst-case complexity. Proof of correctness is also required.

Task 4 [5 Marks]

The previous tasks assume that data is static (drone locations do not change over time). In this task, you are called to rethink your solution of Task 3. More specifically, you are asked to design and implement one single solution as a proof-of-concept application, where **drone locations may change over time**. You need to decide the way changes are applied (e.g., every x seconds y drones change their locations, or every x seconds all drones may potentially change their locations). Your solution should be incremental, i.e., not running everything from scratch. This suggests that you need to somehow track current drone locations and update them accordingly to provide the top-k drone pairs closest to each other.

Task 5 [10 marks]

In this task, you should provide solutions for Task 1, Task 2, Task 3 and Task 4 by using only gen AI tools (e.g., K2Think, ChatGPT, etc.). This task will be submitted separately and at a different time. In addition to the report and code for this task, you should also provide a document explaining the difference between your solution submitted in Phase I and gen AI submitted in Phase II.

4 Deliverables

Submission Phase I

In phase I, you have to submit the required deliverables. Note that **in your report** you should include the following declaration of original work:

> ***I hereby declare that the work presented in the submitted report and the accompanying code is entirely my own. No portion of this submission has been copied, reproduced, or directly generated/refined using the responses or outputs of any AI tools (including, but not limited to, ChatGPT, Copilot, Gemini, DeepSeek, or other automated systems). Any external sources, datasets, or tools that have been used are properly cited and referenced. I understand that any breach of this declaration may result in submission cancellation or significant mark deduction.***

The deliverables of the assignment should include your report and your code. For each task you need to provide:

- Explanation of the baseline and your improved solution, including proof of correctness, worst-case complexity analysis, and execution times in tables and plots with your observations [50% of the task grade].
- Clearly written code with comments/docstrings [50% of the task grade].

Submission Phase II

This submission involves Task 5, which includes the solution to Task 1, Task 2, Task 3 and Task 4 (both report and code) provided solely by the use of gen AI tools. As mentioned in the description of Task 5 previously, you need to provide a report explaining the differences between your solutions and the solutions provided by gen AI for each task.

4.1 Deadlines

The deadline for Submission Phase I is set to Monday, February 9, 2026, 23:59. The deadline for Submission Phase II is set to Friday, Feb 13, 2026, 23:59. **Note: in order for your submission to be valid, both phases must be completed**. Please, pay attention to the rubric shown in the next table. These criteria apply wherever possible and depending on the specific task.

Criterion	Basic	Average	Good	Very Good	Excellent
Problem understanding	Key requirements misunderstood or missing	Main goal understood, some constraints missed	All main requirements understood	Requirements and constraints fully understood	Complete understanding including edge cases
Baseline algorithm	Missing or incorrect approach	Correct but poorly explained	Correct and clearly explained	Well-structured and justified	Clear, precise, and insightful explanation
Improved algorithm	Missing or not better than baseline	Minor improvement, unclear benefit	Clear improvement over baseline	Strong improvement with sound reasoning	Optimal or near-optimal with strong justification
Proof of correctness	Missing or incorrect	Informal, incomplete argument	Mostly correct but lacks rigor	Clear, logically sound proof	Rigorous, well-structured, and convincing
Complexity analysis	Incorrect or missing	Complexity stated without justification	Correct worst-case analysis	Correct and well-explained analysis	Thorough analysis with comparisons
Performance evaluation	No experiments or results	Limited tests, unclear presentation	Tables or plots included	Clear tables and plots with discussion	Insightful analysis of trends and scalability
Top-k variation	Missing or incorrect	Works only for limited cases	Correct for multiple k values	Efficient and well-tested	Elegant, scalable, and well-analyzed
Code correctness	Frequent errors or crashes	Works only for small inputs	Correct for expected inputs	Robust and reliable	Fully robust including edge cases
Code structure	Disorganized, hard to follow	Some structure, inconsistent style	Clear modular structure	Clean, readable, well-organized	Professional-level design
Documentation	No comments or docstrings	Minimal or unclear comments	Adequate comments and docstrings	Clear and helpful documentation	Excellent, concise, instructional documentation
Observations & discussion	Missing or incorrect	Superficial observations	Reasonable interpretation of results	Thoughtful discussion linked to data	Deep insights linked to theory
Gen-AI comparison	Missing	Very shallow comparison	Basic differences identified	Clear, structured comparison	Critical, reflective, and insightful analysis
Academic integrity declaration	Missing	–	–	–	Present and complete

Grade Letter	Quality Points	Default Percentage Equivalent	Grade Description
A+	4.0	95-100	Truly Exceptional Work
A	4.0	95-100	
A-	3.7	90-95	
B+	3.3	87-90	
B	3.0	83-87	
B-	2.7	80-83	
C+	2.3	77-80	
C	2.0	73-77	
C-	1.7	70-73	
D+	1.3	65-70	
D	1.0	60-65	
F	0.0	<60	
WF	0.0	n/a	Withdrawal after the Withdrawal Deadline