# Closest Pair Quest

## Closest Pair Quest – Divide & Conquer

By:

**Daniyal Saeed**

SAP:

**53937**

Instructor:

**Mr. Muhammad Usman Sharif**

Department of Faculty of Computing

**RIPHAH INTERNATIONAL UNIVERSITY,**

**ISLAMABAD, PAKISTAN**

# TABLE OF CONTENT



## Abstract

This project implements a web-based application to find the closest pair of points in a 2D plane. Users can generate a set of random points within a defined canvas and compute the closest pair using two approaches: Brute Force and Divide and Conque

## Introduction

Closest Pair Quest is a simple web-based tool that shows how to find the closest two points among many random points.
It compares:

- Brute Force (slow)
- Divide & Conquer (fast)

The system displays points on the screen and shows the closest pair along with execution time.

## Problem Statement

Students often find it difficult to understand:

- Why brute force becomes slow
- How divide & conquer improves performance
- What the “strip” technique does
- How algorithm complexity affects speed

This tool provides a clear visual demonstration of both methods.

## Objectives

- Visualize closest-pair algorithms
- Compare brute force and divide & conquer
- Show differences in time and performance
- Teach computational geometry interactively
- Create a simple educational web tool

## Tools and Technologies Used

- Python 3 + Flask (backend)
- HTML/CSS/JavaScript (frontend)

## Algorithm Explanation

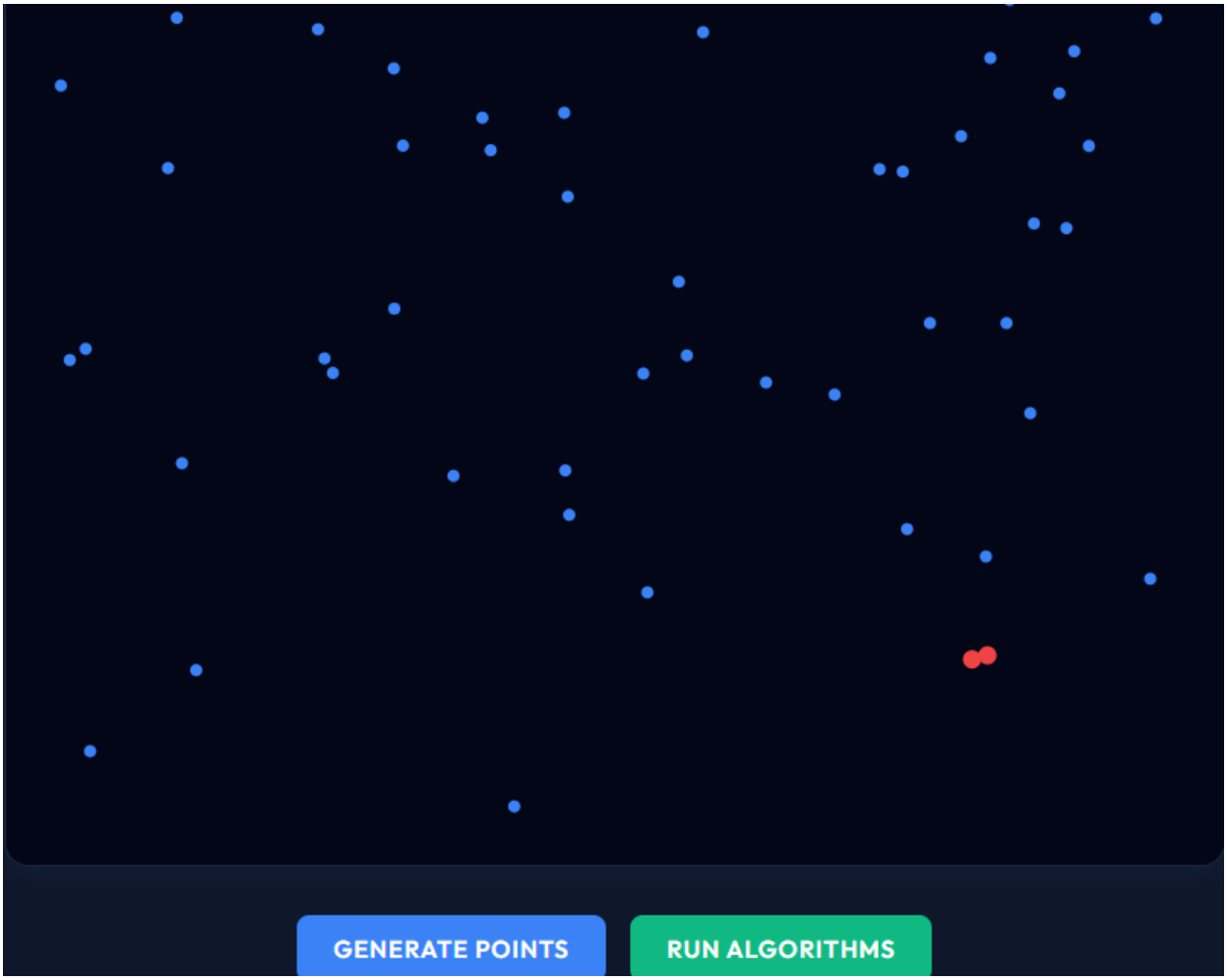
1) **Brute Force**

- Compares every point with every other
- Very easy to understand
- Slow for large number of points

2) **Divide & Conquer**

- Sorts points
- Splits into two halves
- Recursively finds best pair
- Uses strip method for cross-boundary cases
- Much faster and efficient

## User Interface Design




## Time Complexity

| Algorithm | Time Complexity | Explanation |
|---|---|---|
| **Brute Force** | $O(n^2)$ | Checks all pairs |
| **Divide & Conquer** | O(n log n) | Recursive and sorted method |
| **Strip Method** | O(n) | Limited point checks |

## Summary

- Brute Force = slow
- Divide & Conquer = fast

The project visually shows this speed difference

## System Working

1. User selects number of points
2. System generates random points
3. Backend runs both algorithms

## Displays

- Closest pair
- Distance
- Time taken (ms)

## Applications

- AI (nearest-neighbor)
- Robotics (collision detection)
- Maps/GPS (nearest location)
- Computer graphics
- Clustering algorithms

## Limitations

- Works only in 2D
- Large number of points may slow browser
- Basic visualization

## Challenges and Solutions

| Challenge | Solution |
| --- | --- |
| Slow calculation | Use Divide & Conquer |
| Strip logic | Sort strip by Y coordinate |
| Live display | API-based backend |

## Conclusion

Closest Pair Quest is a simple and effective educational project. It helps users understand why divide & conquer is faster than brute force and how geometric algorithms work. The tool gives clear visualization, quick comparison, and easy learning.

## GitHub Link

**https://github.com/daniyalsaeed260**