CPSC-335 Project 2 Project Report and Mathematical Analysis

Group Members:
Michael Bui / buimichael@csu.fullerton.edu
Natalia Garcia / natgarcia@csu.fullerton.edu

Project Report:

This project centers on the design of an algorithm that determines the minimum number of turns required for two knights on a chessboard to capture each other. The knight's position is represented by its x and y coordinates, and a knight can move in one of eight possible L-shaped moves, which would be two squares in one direction and one square in a perpendicular direction. This problem assumes that both knights cooperate to meet as quickly as possible, where each turn allows each to move once. The goal is to calculate the number of turns necessary before one knight lands on the same square as the other. The knights' movements can be represented as vertices and edges on a graph, where the moves of each knight are determined to be connections between two points. The shortest path distance "d" between the two knights can be represented as the fewest number of single-knight moves that is needed for one to reach the other. Both knights can move simultaneously, meaning that the minimum number of turns required would be the ceiling value of half that distance (d/2). This algorithm uses a breadth-first search (BFS) algorithm, which can be used to calculate the shortest path. This results in a time complexity of O(d^2). The sample input knightA at [0,0] and knightB at [4,2] would have the number of turns at 1, which is the minimum number for it, because knightA can move to [2,1] and knightB can capture it in the same turn as knightA.

Mathematical Analysis:

The movement of knights requires a minimum number of turns for two knights to capture each other. Mathematically, their positions as vertices on a graph can be drawn out to where each move is an edge, and the shortest path distance "d" represents the minimum number of single-knight moves. Each turn allows both knights to move once, where the distance between them can shrink by two squares. This would mean that the least number of turns needed before one knight makes contact with another would be "d/2". Both knights can move along the shortest path from opposite ends, where they would meet halfway when d is even or crossing paths in the middle of a turn when d is odd. The knight's distances and determining their movements result in a breadth-first search, where the time complexity of the algorithm is O(d^2) or O(1) when using the closed-form formula.