# *CS194 Project: Sudoku Difficulty*

Kha Chu

Advisor: Professor Sahai

# The Problem

- Sudoku puzzle difficulties tend to not match the actual observed difficulty.
- Estimations of difficulty tend to stem from number of numbers revealed from the start.
- Problem: such a method is innacurate, may not be solvable via pure logic.

# How to Fix the Problem

- Difficulty should be measured in technique.
- Therefore, must analyse solving techniques.
- Should be as “human” as possible.
- Based on how I solve puzzles.

# Outline

- Written in Java, SudokuSolver class:
  - takes a puzzle file as an argument
  - contains solver methods
    - called in succession
    - each call is repeated until no cells are changed, then next method is used

# Methods of Solving

- Each public method is a different method of solving.
- involves:
  - examining a cell's row, column, and block to see if all other values used
  - examining surroundings to see if a value must be in a cell
  - checking surroundings and seeing if any value is forced for a row/column within a block

# Plan

- First quarter: Create the solver that will solve a sudoku puzzle with the tools given.
- Second quarter: Use the solver in conjunction with random number placement to create a puzzle generator that yields a puzzle with a unique solution. Generate a variety of puzzles and survey various people's opinions on the difficulty of the puzzles. Find the correlation between generated difficulty and perceived difficulty.
- Third quarter: Using the results from the survey, tweak the generator and solver in order to increase correlation between generated and perceived difficulty.