# CS375 Project 1

Due Oct 4th 10PM ET

A general note for CS375: When writing up your homework, please write neatly and explain your answers clearly, giving all details needed to make your answers easy to understand. Graders may not award full credit to incomplete or illegible solutions.

As a reminder, projects can be submitted in small groups, optimally with 3 people.

Clear communication **is the point**, on every assignment. In general in CS375, unless explicitly specified otherwise, answers should be accompanied by explanations. Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Necessary acknowledgements: many of these questions are reproductions or adaptations of questions of Eric Aaron's devising.

1. Download the source code for the ArrayIntegerSet class from this page. For each constructor and method in the ArrayIntegerSet class, determine its worstcase time complexity (using $\Theta$ notation) in terms of the number of elements $n$ in the set and, if appropriate, any other data, such as the size $m$ of any parameters. For each method, please add something like the following in its comment header: // Time complexity: $\Theta(n)$, where $n$ is the size of this ArrayIntegerSet.

   For every method, be sure to unambiguously state what each variable in your time complexity expression refers to. For example, if two relevant variables refer to arrays, be very clear which variable refers to which array. (Please be more specific than saying the input array, as from some perspective, any relevant parameter could be viewed as input!)

   Along with each asymptotic complexity bound, include a short (12 sentences at most!) explanation of how you came up with your complexity bound. If your answer depends upon the complexity of other methods in the Class, note that as well. Be sure to give the complexity of all methods (except for main()), whether public or private.

2. On this page you will find 8 different problems: pick 1 from each section (for a total of 3 problems) and write the pseudocode for a brute force algorithm to solve the problem. While detailed correctness arguments are unnecessary, I'm asking for a short English description of what the algorithm does along with a detailed analysis of the worst case runtime.