Analysis of Algorithms
CS 375, Fall 2018
Project 1
Due **AT THE BEGINNING OF CLASS** Wednesday, September 26

Project 1: Determining Algorithmic Time Complexity

In this assignment, you'll work in teams (picked randomly by your prof.) to determine the time complexity of methods in two classes. In Part 1, you will be given source code for the class, and you will figure out the time complexity by analyzing that code. In Part 2, you will *not* be given the source code for the class—you will instead estimate the time complexity by analyzing the number of comparisons the methods make during the sorting process.

The goals of this project are:

- to give you practice determining the worst-case time complexity of algorithms given their source code;
- to give you practice using the scientific method to guess at the worst-case time complexity of algorithms for which you don't have the source code; and
- to give you practice working with another student as a team.

Exercises

1. For **Part 1**, download the source code for the ArrayIntegerSet class from the course Projects and Presentations website.

   For each constructor and method in the ArrayIntegerSet class, determine its worst-case time complexity (in big-Θ 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:  Θ(n), where n is the size of this ArrayIntegerSet
   ```

   You need not explain how you came up with your complexity formula; in particular, you do not need to say what you used as the primitive operation that you counted.

   Be sure to give the complexity of all methods, whether public or private. Also, be sure to describe, as in the sample above, what any variables like n or m refer to.

2. For **Part 2**, download the sortData.csv file from the course Projects and Presentations website. This file contains the data obtained by running 6 sorting algorithms and counting how many comparisons they make during execution. (The sorting algorithms themselves will not be made available to you.) The data in the file was gathered as follows: For each desired array size, a random array A of that size was created; then, each of the 6 sorting algorithms was run on array A. (To be clear: All 6 algorithms were run on the same array A.)

Based on this data, conjecture on the *worst-case* time complexity of each of the 6 algorithms. Then write a report containing these conjectures and explanations supporting them.

Please keep the following notes in mind:

- Please treat this part of the project like a physics or chemistry project in which you are creating a report, making conjectures based on the results of an experiment. Therefore, a significant part of evaluating your work will be the consideration of whether or not the data are sufficiently analyzed or if the conclusions are logical. (In other words, don't just say "Here's our conjecture: ..."; please also include your analysis of the data, such as describing how you came up with your conjecture instead of a different conjecture.)
- As usual in CS375 assignments, clear communication *is* the point: There will be deductions if it is difficult for your grader to understand your process and conclusion. Please treat your written report as being as important as your conclusions.
- Also as usual for CS375 assignments, please feel free to talk with me if you have questions, especially if you are unsure what to include in your report!

What to Hand In

Each group should submit just one set of answers. In particular, each team should submit a folder containing two files:

- For Part 1, turn in a text file that includes only the source code that you analyzed and the time complexity of all the methods. The time complexity should be included in the comment header for each function.
- For Part 2, turn in a PDF file that has your report.

That folder should be named

```
team_<LETTER>_project1
```

where `<LETTER>` is replaced by the letter corresponding to the group in the team assignments. E.g., the folder from *team a* would be called `team_a_project1`.

Submission instructions: Please submit your work to the `Dropbox` folder in the CS375 filespace. (Please note: This is not Moodle!)

- Mount the Courses/CS375 filespace (the same way as you would for other courses).
- Go to the `Course_Materials` directory, then go to the subdirectory called `Dropbox`.
- Copy your folder into the Dropbox. Please make sure to copy the folder as a single item, not copy the two individual files inside the folder!

In addition, each student is required to individually email me within a day after the project was turned in, telling me:

- how well your group worked together;
- how much (percentage) each member of your group contributed to the project; and
- how many hours you (individually) spent on this project.