BitVector Redux

Overview

The goal of your second programming assignment is to improve the implementation of `BitVector` that you just completed. In particular, you are to optimize implementations for the two use cases described in PA1, you should also write a comprehensive set of unit tests, some stress tests and document your code.

Details

The test cases must be written using the Junit framework with the template provided on the PA2 web page. You should implement a set of automated tests that following the example given in the template in `src/test`.

Documentation should be written using JavaDoc and should describe your implementation at a reasonable level of detail – that is to say sufficiently clearly such that someone else may understand the big picture of the algorithms used in the implementation.

You may have to provide several implementations of BitVector tuned to the different use cases.

Requirements

For PA2 we expect you to turn in several Java classes. Please beware that you follow the naming convention, failing to do so may prevent our grading programs from processing your submission. All `.java` files in your assignment should be named `ClassName_uid.java` where `uid` stands for an identifier that will be given by the TA. There must at least be one file named `Director_uid.java` containing a `public` class of the same name, with a `public` empty constructor taking no arguments. All classes must be in the `cs307.first` package.

You testing classes should be in a subpackage with your uid, e.g. `test.first.uid`.

Furthermore we expect that you will turn in a PDF file that contains a short development log accounting the time you spent on different tasks: design, implementation, testing, debugging, documentation. The report should also contain any comments on the assignment, open questions, etc.

Use `turnin -c cs307 -p pa2 filelist` to submit the assignment.