Implementation and Analysis					
	Exemplary A	Good B	Satisfactory C	Marginal D	Unacceptable F
Adherence to Specifications (25 points)	There are no deviations from the specifications. Items are named correctly. Files are named correctly and all files are present.	Most components adhere to specifications. Some issues remain. Examples may include spelling or case of specified names or other minor issues.	Many components meet specifications, but key elements are incorrect. Examples include algorithmic deviations from what we covered in class, or use of built-in Python features instead of custom implementations.	The submission attempts to follow instructions, but fundamental issues persist. There is a clear lack of understanding of the specifications and how they relate to the material.	The submission may be substantially incomplete. It may be missing specified components. It may implement items in ways other than specified or covered. The submission conveys ignorance or illiteracy of specifications.
Functionality (100 points) autograded	The submission passes all test cases. Output is formatted correctly, with no unspecified output. Main code conditioned to be separate from definitions.	The submission produces the correct result, but minor issues such as formatting remain.	There is evidence that the implementation does not consider problematic cases or special cases. The program may crash on a few test cases.	The implementation is fundamentally incorrect. Multiple cases fail, or the program crashes in multiple scenarios. The submission fails to convey sufficient understanding.	The program crashes on most inputs. It contains errors that imply it could not have run during development. Of the cases that do not crash, many fail with incorrect results.
Performance (25 points) and Analysis (20 points)	The implementation takes advantage of all opportunities to improve performance. Performance analyses are presented and justified correctly.	The implementation is generally efficient, but some steps could be taken to improve performance. Stated performance is correct, but justification is lacking.	The implementation misses many opportunities to improve performance. Steps are taken without regard to their cost. Analysis is incorrectly stated or justification is very weak.	The implementation is naïve with respect to performance. Some sections of code take significantly longer than the algorithms described in class. Neither the performance analysis nor its justification is correct.	The run time interferes with grading. it contains unnecessary loops and/or repetition. If sections of code perform at all, they are extremely inefficient. No performance analysis is presented.
Readability (10 points)	Variable, function, and class names are descriptive. Comments explain sections of code where necessary. Indentation is consistent and deliberate.	Code is well-structured, but names could be more descriptive. Some sections of code are not obvious and could benefit from additional comments.	There is repeated code that should be moved to functions. Names are not reflective of what they reference. Inconsistent white space may be present. Unnecessary special cases are implemented.	Code uses inappropriate features. Names are poorly chosen. Complex implementations are not commented. Repeated code degrades reliability. Functions return values inappropriately.	The implementation conveys a fundamental misunderstanding of basic programming structures. Loops and conditionals are interchanged. Incorrect variables are used for computation. Sections of code are unreachable.
Testing (20 points)	Incorporates all cases needed to fully test implementation. Follows examples, including proper formatting.	Incorporates nearly all necessary cases for full testing. Follows examples, including proper formatting.	Incorporates only an incomplete subset of cases. Purpose of some tests may not be clear. Shows lack of consideration of provided examples.	Missing many necessary. Purpose of tests is not clear, does not follow examples.	Testing is largely incomplete and missing most necessary cases. Tests follow no logical structure or order.