

# Algorithm Write-up Guidelines

Throughout the class, you will be asked to write-up an algorithm you have designed to solve a problem. Whether on a homework assignment, project, or an exam, your solution is expected to contain the following sections. Each section should be long enough to contain a full, readable explanation of your solution, and no longer. You may assume that the reader is familiar with the problem.

- **Description.** An English description of your algorithm, similar to what you might say out-loud when asked to describe how it works in class. It might be useful to include an example to illustrate your approach.
- **Pseudocode.** A pseudocode description of your algorithm, which does not rely on a specific programming language's syntax, but instead uses English phrases where appropriate. For example, "insert x at the end of the list" is much clearer than "list.insertAtEnd(x)." The pseudocode should give sufficient detail to make the analysis straightforward (e.g. don't hide for loops in an English phrase) while remaining at a high enough level so that it can be easily read and understood. Be sure that the interpretation of your pseudocode is unambiguous and that you include explicitly the input and output values.
- **Time Analysis.** A worst-case analysis of your algorithm's running time. This is where you should explicitly state the data structures that are being assumed and how their preprocessing and/or query times affect your analysis.
- **Proof of Correctness.** A proof that your algorithm does what you say it does and is optimal (if appropriate). If this is a proof by induction, be sure to state clearly what the base case, induction hypothesis, and induction steps are. If this is a proof by contradiction, state clearly what the assumption to be contradicted is. In general, be sure to define any terms used and construct a logical argument using complete mathematical sentences.