Analysis of Nonrecursive algorithms

- Read 2.3, pages 61-66
    - Consider the algorithm described in Example 1, for finding the largest element in a list of n numbers.
        - Apart from the array, what other information does the algorithm need to store in memory, i.e. how much more memory does it need to use?
        - Explain what the "basic operation" is in this case, and how we arrived at that conclusion.
        - Explain how we count how many basic operations are needed in this algorithm. What is the algorithm's time efficiency class?
    - Describe the five steps of the general plan for analyzing the time efficiency of a non-recursive algorithm, and identify those steps in the earlier example.
    - Review Appendix A (pages 475-577). You do not need to memorize those formulas, but you should familiarize yourself with them and make sure you understand what each says.
    - Study the algorithm described in Example 2, for determining if an array of elements contains two equal elements.
        - In what substantial way does this example differ from the first example?
        - Explain how the loop index limits make sense.
        - Explain what the "basic operation" is in this case.
        - Do we need to distinguish between worst-case, best-case and average-case in this instance? Why, or why not?
        - Follow the math steps at the top of page 64. Compare the steps used with formulas from the appendix.
    - Using appendix A where needed, answer the exercises:
        - 2.3.1a, 2.3.1b, 2.3.1d, 2.3.2a, 2.3.4