Brooks Jackson

3.3, 10.1 Journal

1. Review the code for each recursive program binarySearch() on page 147 and answer the following questions.
    a. What is the greatest similarity to any recursive program we have investigated so far? Briefly explain you answer.

    The greatest similarity is that there is the presence of a base case that determines when the recursion should stop.

    b. What is the greatest difference to any recursive program we have investigated so far? Briefly explain you answer.

    The greatest difference is that the array divides into halves each step, whereas other recursive problems are typically divided into smaller subproblems of the same kind.

    c. Identify the "base case" as described in the Key Concepts box on page 116.

    The base case is when the value is being searched ("value == anArray[mid]"

    d. In your own words, describe why binarySearch() is an effective solution for the search problem?

    Because it takes advantage of the fact that the array is sorted. It divides the array in half so it eliminates the need to search the entire array.

2. In Section 10.1, select a sentence that makes the best case for measuring the efficiency of algorithms. Briefly explain why you chose this passage.

"The particular operations that the algorithms require can cause A1 to run faster than A2 on one computer, while the opposite is true on another computer." I like this passage because it shows how important measuring efficiency independent is. It also shows that the focus should be on understanding the efficiency rather than comparing implementations.

3. Algorithm growth rates are described on pages 467-475. In no more than one paragraph, summarize the section. What passage did you find most confusing?

The section on algorithm growth rates describes the importance of understanding a growth rate of an algorithm's time requirement as a function. It emphasizes that doing so allows for assessing the efficiency independently of other factors like implementations and computer systems. They also mention that the efficiency is mainly a concern for large problems, and time requirements for small problems don't really matter. One part that was confusing was "Figure 10-1 also shows that A's time requirement does not exceed B's until n exceeds 25." It was confusing because it doesn't explicitly explain what A and B represent or their time requirement measurements.