

Problem 2: Binary Search

- a. Write pseudocode for binary search in CLRS notation that uses a while loop.
- b. Identify the run time of this algorithm by counting operations (show your work), and prove the algorithm's big O.

- c. Prove that binary search solves the search problem on a sorted array by using a loop invariant proof.

Problem 3: Insertion sort using binary search

- a. Look at the pseudocode on page 19 of CLRS for Insertion-Sort. Line 5 includes a while loop that performs a backward linear search to identify the correct position for the key. Rewrite the pseudocode to perform a backwards binary search at this point in the algorithm.

- c. Write a proof of correctness showing that this modification still solves the sorting problem, using loop invariants. Each loop will need its own loop invariant, as well as discussion of that loop invariant's state for initialization, maintenance, and termination.

Problem 4: Big and Small sort

- a) Write CLRS pseudocode for an algorithm that solves the sorting problem by iterating through the unsorted array and finding both the smallest and the largest item in the array in each pass through the array, and placing them both in the correct position in the array. Make this algorithm as efficient as possible.
- b) Determine the big O runtime of this algorithm by counting operations (show your work). Prove its big O using the definition of big O.

- c) Prove that this algorithm solves the sorting problem using a loop invariant proof. Each loop will need its own loop invariant, as well as discussion of that loop invariant's state for initialization, maintenance, and termination.