

Design and Analysis of Algorithms

Problem#1

Consider the following variation of insertion sort. Assume A is the array we are sorting and that $A[1], A[2], \dots, A[p]$ is the subarray of A which is currently sorted. Rather than sequentially searching for $A[p + 1]$'s position in $A[1], \dots, A[p]$, suppose we perform a binary search. If this is the case for $0 < p < n$ (where n is the size of A), then what is the time complexity of the sorting algorithm? Explain.

Problem #2

Propose at least two strategies other than chaining for handling *collisions* in a hash table. Give their advantages and disadvantages.

Problem#3

1. Give definition of a heap.
2. What minimal sequence of *insert* and/or *removeMin* operations on heap A will transform it into heap B ? Draw the heap after each operation.

