

Design and Analysis of Algorithms

Problem#1

Describe an algorithm that, given n integers in the range 0 to k , preprocesses its input and then answers any query about how many of the n integers fall into range $[a..b]$ in $O(1)$ time. Your algorithm should use $O(n+k)$ preprocessing time. Separate clearly the preprocessing part from the query answering part in your algorithm. Analyze the run-time of each part.

Problem #2

What is the difference between the binary-search-tree key property and the min-heap key property? Can the binary-search-tree key property be used to print out the keys of an n -node tree in sorted order in $O(n)$ time? Can the min-heap key property be used to print out the keys of an n -node tree in sorted order in $O(n)$ time? Explain how or why not.

Problem#3

Describe an efficient algorithm that, given set $\{x_1, \dots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contain all of the given points. Argue that your algorithm is correct.