Course based on MIT 6.046J & Clemson CPSC 840
Soliton Technologies, Bangalore

Design and Analysis of Algorithms

Problem Set: 2 Due Date: Sept 28, 12:00am
Instructor: Sumod K Mohan Schedule : MWTh 5:30pm-7:00pm

1.1 Median of two sorted Arrays : 15 Pts

You are given two sorted arrays $A[1....n/2]$ and $B[1...n/2]$ as input. Please show how to computer the median among all the n elements contained in both arrays in $O(n \log n$ time.

1.2 Missing Element : 15 Pts

You are given two arrays: A, containing n elements, and B, containing n-1 elements present in A. Design a comparison-based algorithm running in $O(n)$ time that determines which element in A does not appear in B.

1.3 Fun with Intervals : 30 Pts

You are given n intervals $[a_1, b_1]....[a_n, b_n]$ on the number line. For simplicity, assume all interval endpoints are distinct. Two intervals can be related in 3 possible ways : they can *nest* (one interval within the other), they can be *disjoint* (not overlapping at all), or they can *cross* (overlapping but not nesting). A set of intervals is said to be *laminar* if there are no crossing pairs of intervals; that is, if two intervals overlap at all, they must nest. Laminar intervals exhibit a sort of "balanced parenthesis" structure. Give an $O(n \log n)$ algorithm for testing whether or not a set of n intervals are laminar.

1.4 Prog: Another Foray into STL. 10 pts

A graph is a mathematical data structure consisting of node and edges connecting them. To help visualize it, you can think of it as a map of "cities" (*nodes*) and "roads" (*edges*) connecting them. In a *directed graph*, the direction of edges matter, that is a edge from A to B is not also an edge from B to A. One way to represent a graph, is by assigning each node a unique ID number. Then, for each node ID n, you can store a list of node ID's to which n has an outgoing edge. This list is called an *adjacency list.*

Write a Graph Class that uses STL containers (vectors, maps etc.) to represent a directed graph. Each node should be represented by a unique integer (an int). Provide the following member functions:

- `Graph::Graph(const vector &starts, const vector &ends)`

  Constructs a *Graph* with given set of edges, where *starts* and *ends* represent the ordered list of edges' start and endpoints.

- `int Graph::numOutgoing(const int nodeID) const`

  Returns the number of outgoing edges from nodeID - that is, edges with *nodeID* as the start point.

- `const vector<int> &Graph::adjacent(const int nodeID) const`

  Returns a reference to the list of nodes to which nodeID has outgoing edges.

1.5 Prog: From Hacker to Master : 30 Pts

Perfection is achieved, not when there is nothing more to add, but when there is nothing to remove.

As with any skill, clean problem solving and programming skill requires diligent hard work. Hackerrank is a website that provides problems at various levels of complexity and has an automated code checker. So let us get started on this journey. Please create a user account on it and do the following problems.

1. **Filling Jars** problem in the Algorithms Track under Warm-up category.
2. **Mark and Toys** problem in the Algorithms Track under Arrays and Sorting.

PS: For all the programming problems, please use the appropriate STL containers.