Sample Problem Set for Exam

COSC 2430: Programming and Data Structures

Instructor: Arjun Mukherjee

IMP Note: These are sample problems and are of the similar difficulty level you may expect in the exam. Hence, solutions are purposely not provided to encourage the student to solve and have the TA evaluate.

(a). What is the difference between shallow and deep equality tests on Arrays in Java?

(b) How would you backup (copy) all elements of an array to a new array using a *single* Java statement?

(c) Provide an algorithm/pseudocode for finding the penultimate (second-last) node in a doubly linked list where the last node is indicated by a null next reference.

(d) Provide an algorithm/pseudocode to find the k^{th} last element of a singly linked list starting with only the header sentinel?

(e) Which function has the similar profile (i.e., the same "shape") in the log-log scale as it is in the classical y/x scale?

(f) How does one order these functions based on non-increasing asymptotic growth rates? $5n^{2.5}, 2^{logn+5n}, 3^{nlogn}, n^2 + 10logn, 3logn + 4n^{log5n}$

(g) Textbook Exercises R-5.3, R-5.4, R-5.9

(h) Textbook Exercise C-5.20 Hint: Leverage the fact that whether the last element is odd or even can be checked by the modulo operator (%) in Java and then based on this recur.

(i) Textbook Exercise R-6.3, R-6.5, R-6.9

(j) Textbook Exercise R-8.20

(k) When do collisions occur?

(l) What are two good collision handling shcmes?

(m) What is open addressing? What is the basic assumption for it to work?

(n) What is the downside of separate chaining?

(o) What modifications are needed for the hash to work with linear probing?

(p) Outline key steps in entry deletion under linear probing.

(q) Draw the 11-entry hash table that results from using the hash function, h(i) = (4i+7) mod 11, to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by chaining.

(r) What are worst case running times for insert, min and removeMin methods for a priority queue using (a) sorted list (b) unsorted list implementation.

(s) Textbook Exercise R-9.4, R-9.7, R-9.9, R-9.11

(t) Textbook Exercise R-11.1

(u) Insert, into an empty binary search tree, entries with keys 21, 0, 4, 8, 38, 206, 101, 11, 3 (in this order). Draw the tree after each insertion.

(v) Textbook Exercise R-12.20, R-12.23

(w) What are runtimes of merge and quick sort in in n input sequence? When is one better than the other? When are they both equally competitive?

(x) Briefly define edge list, adjacency list, adjacency map and adjacency matrix representations of a Graph. What would you prefer if the graph edge set is very sparse and it contains a larger set of nodes? Why?

(y) Define back, cross, and forward edges for BFS on an undirected graph. How do they differ from an DFE search tree?

(z) What is a minimum spanning tree? How do Prim Jarnik and Kruskal's methods differ in their execution. Which kind of method would you prefer for what kinds of graphs and why?