Computer Science 252
Algorithms and Data Structures

Practice Problems

Important note: These questions do not mirror the exact topics covered on the exam, but instead attempt to give a feel for what questions may resemble on the exam. This set of problems is not comprehensive; some topics that were covered in class are *not* covered in this set of practice problems but *may still appear on the exam.* Likewise, topics covered here not necessarily on the exam.

Written by William Zahary
williamzahary@gmail.com

McGill University
COMP 252B, 2023
Taught by Dr. Luc Devroye

Algorithm Theory

1. MASTER THEOREM. State, in Θ notation, the complexities of the following recurrences using Master theorem. If Master theorem cannot be used, state why.

   (a) $3T_{n/3} + \sqrt{n}$

   (b) $2T_{n/2} + \frac{n}{\log_2(n)}$

   (c) $7T_{n/3} + n^2$

2. DYNAMIC PROGRAMMING. Consider a modified version of the Knapsack problem, where, as per usual, we are given N items with integer sizes $x_1, \ldots, x_N$ and a knapsack of size K. *Unlike the typical problem*, we are now able to choose one item to "cut" into two pieces, one of which must be at least 75% the size of the initial size, and we may use one of the two pieces. In other words, for some item j, we can let its new weight be y_j such that $y_j \in (0, 0.25x_j] \cup [0.75x_j, x_j]$. We want to determine if there exists a subset $S \subseteq \{1, \ldots, N\}$ and specific item $j \notin S$ such that $y_j + \sum_{i \in S} x_i = K$.

   (a) How must we adjust the Knapsack algorithm studied in class to solve this new problem?

   (b) Provide your algorithm for computing whether the set $S \cup \{j\}$ exists.

   (c) What is the time complexity for this problem?

3. DECISION TREES. Prove by induction that any k-ary decision tree with L leaves has height $h \geq \log_k L$.

Data Structures

4. BINARY TREES.

   (a) How many binary trees can be constructed on n nodes?

   (b) Briefly provide some motivation for how this conclusion could be reached.

   (c) What does this say about the number of bits required to store a binary tree containing n nodes linearly?

5. RED-BLACK TREES.

   (a) A valid red-black tree has n internal nodes. How many external nodes does it have? How many edges does it have?

   (b) If the rank of the root of a valid red-black tree with n internal nodes is 7, what can you deduce about n?

(c) Describe or draw a situation where a ROTATION would occur on a red-black tree.

6. AUGMENTED DATA STRUCTURES. In class, we discussed the hypothetical problem of checking whether any two rectangles overlap in a set of rectangles, knowing the four corners of each rectangle in the form of coordinates. Briefly explain how you could use an augmented data structure to solve this problem in $O(n \log n)$, and explain how you would build the augmented data structure.

7. PRIORITY QUEUES. Recall that the binary heap and the tournament tree are two data structures that implement priority queues, and that we can sort a list of elements using priority queues.

   (a) In pseudocode, describe the algorithm for HEAPSORT on a list of n elements.

   (b) In pseudocode, describe the algorithm for TOURNAMENT TREE SORT on a list of n elements.

   (c) Although both sorting algorithms have $\Theta(n \log n)$ complexity, one uses much fewer comparisons than the other and is thus much closer to the lower bound. Which one is it?

8. STRINGOLOGY. We are given the string $T = abcabcabd$ and the pattern $P = abcabd$.

   (a) Draw the suffix trie for T.

   (b) Make the magic table for P.

   (c) Explain the steps the Knuth-Morris-Pratt algorithm would take to verify if P occurs in T, using the magic table.

Information Theory

9. SHANNON'S THEORY.

   (a) What is meant by "binary entropy" ($\mathcal{E}$)?

   (b) State Kraft's inequality.

10. LEMPEL-ZIV COMPRESSION.

   (a) Using the Lempel-Ziv compression method, how many bits are required to encode a symbol for an alphabet of size k?

   (b) Draw the digital search tree for the string $T = bbabcabccaa$.

11. HUFFMAN TREES. Construct the Huffman tree for the table of probabilities given below, and list the corresponding codes.

Input	Probability
a	0.3
b	0.04
c	0.16
d	0.21
e	0.23
f	0.06

Graphs

12. BREADTH-FIRST SEARCH. Suppose we are given a graph $G = (V, E)$.

    (a) Provide an algorithm that visits every node of G in breadth-first order and state its complexity.

    (b) Write an $O(|V| + |E|)$ algorithm to determine whether G is a bipartite graph[1].

13. EULER GRAPHS.

    (a) When is an undirected graph Eulerian?

    (b) When is a directed graph Eulerian?

    (c) Give, in psuedocode, the algorithm for finding an Euler tour in a connected graph G that is known to be Eulerian. Assume we are given G and some start node.

14. DAGS.

    (a) Give a succinct algorithm for constructing the graph of all strongly connected components of a directed graph (though not acyclic) G.

    (b) Is the resulting graph necessarily a dag? If yes, provide a proof. If not, give a counterexample.

15. DIJKSTRA'S ALGORITHM.

    (a) What is the time complexity of Dijkstra's algorithm using a binary heap? Using properties of the algorithm, give your reasoning.

    (b) What is the time complexity of Dijkstra's algorithm using an array? Again, give your reasoning.

[1] Recall a graph is *bipartite* if its vertex set V can be partitioned into two disjoint sets $V = A \cup B$ such that no two vertices in A share an edge, and no two vertices in B share an edge.