HW2

2024-10379

June 19 2025

1 Chapter 4

1. 4-12) Devise an algorithm for finding the k smallest elements of an unsorted set of n integers in $O(n + k \log n)$.
2. 4-14) Give an $O(n \log k)$-time algorithm that merges k sorted lists with a total of n elements into one sorted list. (Hint: use a heap to speed up the elementary $O(kn)$-time algorithm)
3. 4-16) Use the partitioning idea of quicksort to give an algorithm that finds the median element of an array of n integers in expected $O(n)$ time. (Hint: must you look at both sides of the partition?)
4. 4-20) Give an efficient algorithm to rearrange an array of n keys so that all the negative keys precede all the nonnegative keys. Your algorithm must be in-place, meaning you cannot allocate another array to temporarily hold the items. How fast is your algorithm?
5. 4-35) Let M be an $n \times m$ integer matrix in which the entries of each row are sorted in increasing order (from left to right) and the entries in each column are in increasing order (from top to bottom). Give an efficient algorithm to find the position of an integer x in M, or to determine that x is not there. How many comparisons of x with matrix entries does your algorithm use in worst case?

2 Chapter 5

1. 5-3) PRove by induction that there is a unique path between any pair of vertices in a tree.
2. 5-5) Give a linear algorithm to compute the chromatic number of graphs where each vertex has degree at most 2. Must such graphs be bipartite?
3. 5-9) Suppose an arithmetic expression is given as a tree. Each leaf is an integer and each internal node is one of the standard arithmetical operations $(+, -, *, /)$. For example, the expression $2 + 3 * 4 + (3 * 4)/5$

is represented by the tree in Figure 5.17(a). Give an $O(n)$ algorithm for evaluating such an expression, where there are n nodes in the tree.

4. 5-10) Suppose an arithmetic expression is given as a DAG (directed acyclic graph) with common subexpressions removed. Each leaf is an integer and each internal node is one of the standard arithmetical operations $(+, -, *, /)$. For example, the expression $2+3*4+(3*4)/5$ is represented by the DAG in Figure 5.17(b). Give an $O(n+m)$ algorithm for evaluating such a DAG, where there are n nodes and m edges in the DAG. Hint: modify an algorithm for the tree case to achieve the desired efficiency.

5. 5-14) A vertex cover of a graph $G = (V, E)$ is a subset of vertices $V' \in V$ such that every edge in E contains at least one vertex from V'. Delete all the leaves from any depth-first search tree of G. Must the remaining vertices form a vertex cover of G? Give a proof or a counterexample.

3 Chapter 6

1. 6-2) Is the path between two vertices in a minimum spanning tree necessarily a shortest path between the two vertices in the full graph? Give a proof or a counterexample.

2. 6-3) Assume that all edges in the graph have distinct edge weights (i.e., no pair of edges have the same weight). Is the path between a pair of vertices in a minimum spanning tree necessarily a shortest path between the two vertices in the full graph? Give a proof or a counterexample.

3. 6-15) Let $G = (V, E)$ be an undirected weighted graph, and let T be the shortest-path spanning tree rooted at a vertex v. Suppose now that all the edge weights in G are increased by a constant number k. Is T still the shortest-path spanning tree from v?

4. 6-20) Can we modify Dijkstra's algorithm to solve the single-source longest path problem by changing minimum to maximum? If so, then prove your algorithm correct. If not, then provide a counterexample.

5. 6-23) Arbitrage is the use of discrepancies in currency-exchange rates to make a profit. For example, there may be a small window of time during which 1 U.S. dollar buys 0.75 British pounds, 1 British pound buys 2 Australian dollars, and 1 Australian dollar buys 0.70 U.S. dollars. At such a time, a smart trader can trade one U.S. dollar and end up with $0.75 \times 2 \times 0.7 = 1.05$ U.S. dollars — a profit of 5

   $R_{[c_1,c_{i1}]} \cdot R_{[c_{i1},c_{i2}]} \cdots R_{[c_{ik-1},c_{ik}]} \cdot R_{[c_{ik},c_1]}$

   Hint: think all-pairs shortest path.