

CSE5311 Design and Analysis of Algorithms
Exercise Problems 2
09/09/09

1. You are given a list of numbers for which you need to construct a min-heap. (A Min-heap is a binary tree in which every key is less than or equal to the keys in its children.) Write a min-heap algorithm and analyze its complexity.
2. Explain how we can check a graph's cyclicity by using Breadth-first search. Does either of the two traversals – DFS or BFS –always find a cycle faster than the other? If your answer is yes, indicate which of them is better and explain why it is the case; if you answer no, give two examples supporting your answer.
3. Let $G = (V, E)$ be an (undirected) graph with costs $c_e \geq 0$ on the edges $e \in E$. Assume you are given a minimum-cost spanning tree T in G . Now assume that a new edge is added to G , connecting two nodes $v, w \in V$ with cost c .
 - a. Give an efficient algorithm to test if T remains the minimum-cost spanning tree with the new edge added to G (but not to the tree T). Make your algorithm run in time $O(|E|)$. Can you do it in $O(|V|)$ time?
 - b. Suppose T is no longer the minimum-cost spanning tree. Give a linear-time algorithm (time $O(|E|)$) to update the tree T to the new minimum-cost spanning tree.
4. Let $G = (V, E)$ be a connected undirected weighted graph. Assume for simplicity that the weights are positive and distinct. Let e be an edge of G . Denote by $T(e)$ the spanning tree of G that has minimum cost among all spanning trees of G that contains e . Design an algorithm to find $T(e)$ for all edges $e \in E$. The algorithm should run in time $O(|V|^2)$.
5. An Euler circuit of an undirected graph $G(V, E)$ is a path that starts and ends at the same node and contains each edge of G exactly once.
 - a. Show that a connected, undirected graph has an Euler circuit if and only if each node is of even degree.
 - b. Let $G(V, E)$ be an undirected graph with m edges in which every node is of even degree. Give an $O(|V|)$ algorithm to construct an Euler circuit for G .
6. Let $G = (V, E)$ be a directed graph and let v and w be two vertices in G . Design a linear time algorithm to find the number of different shortest paths (not necessarily vertex disjoint) between v and w . The path can be in terms of the number of edges.
7. Let $G = (V, E)$ be a directed weighted graph such that all weights are positive. Let v and w be two vertices of G and $k \leq |V|$ be an integer. Design an algorithm to find the shortest path from v to w that contains exactly k edges. The path need not be simple.
8. Show that the number of different paths of length $k > 0$ from the i th vertex to the j th vertex of a graph (undirected or directed) equals the (i, j) th element of A^k , where A is the adjacency matrix of the graph.

9. A child wants to construct the tallest tower possible out of building blocks. She has n types of blocks, and an unlimited supply of blocks of each type. Each type- i block is a rectangular solid with linear dimensions $\langle x_i, y_i, z_i \rangle$. A Block can be oriented so that any two of its three dimensions determine the dimensions of base and the other dimension is the height. In building a tower one block may be placed on top of another block as long as the two dimensions of the base of the upper block are each strictly smaller than the corresponding base dimensions of the lower block. (Blocks oriented to have equal-sized bases can not be stacked). Design an efficient algorithm to determine the tallest tower the child can build.
10. Design an algorithm to find a vertex in a connected undirected graph whose removal does not disconnect the graph. The algorithm should run in linear time.
11. Let $G = (V, E)$ be a connected weighted undirected graph, and let T be an MCST of G . Suppose that we now add a new vertex v to G , together with weighted edges from v to vertices of G . Design a linear-time algorithm to find a new MCST that includes v .
12. Suppose that you are given an undirected graph with a list of all the articulation points. Show how to find the biconnected components without resorting to running the whole biconnected component algorithm.
13. A Hamiltonian path is a simple path includes all the vertices in the graph. Design an algorithm to determine whether a given acyclic directed graph $G = (V, E)$ contains a Hamiltonian path. The algorithm should run in linear time.