آنالیز الگوریتم‌ها (۲۲۸۹۱)

[بهار ۹۹]

تمرین‌های غیرتحویلی از کتاب CLRS

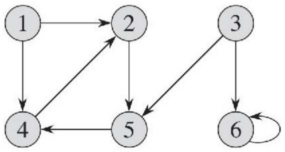
---

1. (22.1 − 5)The square of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$ such that$(u, v) \in E^2$ if and only G contains a path with at most two edges between u and v. Describe efficient algorithms for computing G^2 from G for both the adjacency- list and adjacency-matrix representations of G. Analyze the running times of your algorithms.

2. (22.1 − 6) Most graph algorithms that take an adjacency-matrix representation as input re- quire time $\Omega(V^2)$, but there are some exceptions. Show how to determine whether a directed graph G contains a universal sink—a vertex with in-degree $|V| - 1$ and out-degree 0—in time $O(V)$, given an adjacency matrix for G.

3. (22.1 − 8) Suppose that instead of a linked list, each array entry Adj[u] is a hash table containing the vertices v for which $(u, v) \in E$. If all edge lookups are equally likely, what is the expected time to determine whether an edge is in the graph? What disadvantages does this scheme have? Suggest an alternate data structure for each edge list that solves these problems. Does your alternative have disadvantages compared to the hash table?

4. (22.2 − 1) Show the d and π values that result from running breadth-first search on the directed graph of Figure 22.2(a), using vertex 3 as the source.



(a)

5. (22.2 − 7) There are two types of professional wrestlers: “babyfaces” (“good guys”) and “heels” (“bad guys”). Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have n professional wrestlers and we have a list of r pairs of wrestlers for which there are rivalries. Give an $O(n + r)$-time algorithm that determines whether it is possible to designate some of the wrestlers as babyfaces and the remainder as heels such that each rivalry is between a babyface and a heel. If it is possible to perform such a designation, your algorithm should produce it.

6. (33.4−3) We can define the distance between two points in ways other than euclidean. In the plane, the $L_m-distance$ between points p_1 and p_2 is given by the expression $(|w_1 - w_2|^m + |y_1 - y_2|^m)^{1/m}$.

Euclidean distance, therefore, is $L_2 - distance$. Modify the closest-pair algorithm to use the $L_1 - distance$, which is also known as the Manhattan distance.

7. (22.3 – 1) Make a 3-by-3 chart with row and column labels WHITE , GRAY , and BLACK . In each cell (i, j) , indicate whether, at any point during a depth-first search of a directed graph, there can be an edge from a vertex of color i to a vertex of color j . For each possible edge, indicate what edge types it can be. Make a second such chart for depth-first search of an undirected graph.

8. (22.3 – 2) Show how depth-first search works on the graph of Figure 22.6. Assume that the for loop of lines 5–7 of the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.

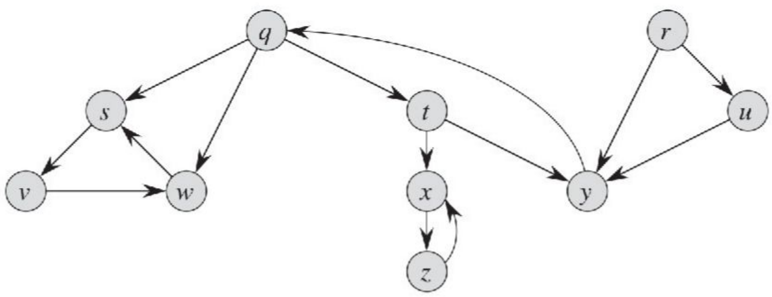


Figure 22.6 A directed graph for use in Exercises 22.3-2 and 22.5-2.

9. (22.3 – 5) Show that edge (u, v) is

   a. a tree edge or forward edge if and only if $u.d < v.d < v.f < u.f$,

   b. a back edge if and only if $v.d \leq u.d < u.f \leq v.f$, and

   c. a cross edge if and only if $v.d < v.f < u.d < u.f$.

10. (22.3 – 8) Give a counterexample to the conjecture that if a directed graph G contains a path from $utov$, and if $u.d < v.d$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced.

11. (22.3 – 9) Give a counterexample to the conjecture that if a directed graph G contains a path from u to v, then any depth-first search must result in $v.d \leq u.f$.

12. (22.3 – 11) Explain how a vertex u of a directed graph can end up in a depth-first tree containing only u, even though u has both incoming and outgoing edges in G.

13. (22.4 – 4) Prove or disprove: If a directed graph G contains cycles, then $TOPOLOGICAL - SORT(G)$ produces a vertex ordering that minimizes the number of "bad" edges that are inconsistent with the ordering produced.

14. (16.4 – 1) Show that (S, I_k) is a matroid, where S is any finite set and I_k is the set of all subsets of S of size at most k, where $k \leq |S|$.

15. $(23.1-1)$ Let (u, v) be a minimum-weight edge in a connected graph G. Show that (u, v) belongs to some minimum spanning tree of G.

16. $(23.1-2)$ Professor Sabatier conjectures the following converse of Theorem 23.1. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A, and let (u, v) be a safe edge for A crossing $(S, V - S)$. Then, (u, v) is a light edge for the cut. Show that the professor's conjecture is incorrect by giving a counterexample

17. $(23.1-3)$ Show that if an edge (u, v) is contained in some minimum spanning tree, then it is a light edge crossing some cut of the graph.

18. $(23.1-8)$ Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T . Show that for any other minimum spanning tree $T^{'}$ of G, the list L is also the sorted list of edge weights of $T^{'}$.

19. Problem $15-1$ Longest simple path in a directed acyclic graph
    Suppose that we are given a directed acyclic graph $G = (V, E)$ with real-valued edge weights and two distinguished vertices s and t. Describe a dynamic programming approach for finding a longest weighted simple path from s to t. What does the subproblem graph look like? What is the efficiency of your algorithm?

20. Problem$16-1$ Coin changing
    Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer.

    a. Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution.
    b. Suppose that the available coins are in the denominations that are powers of c, i.e., the denominations are $c^0, c^1, ..., c^k$ for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.
    c. Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every value of n.
    d. Give an $O(nk)$-time algorithm that makes change for any set of k different coin denominations, assuming that one of the coins is a penny.

21. $(24.1-4)$ Modify the Bellman-Ford algorithm so that it sets $v.d$ to $-\infty$ for all vertices v for which there is a negative-weight cycle on some path from the source to v.

22. $(24.1-5)$ Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \to R$. Give an $O(VE)$-time algorithm to find, for each vertex $v \in V$, the value $\delta^*(v) = \min_{u \in V}\{\delta(u, v)\}$.

23. $(24.2-4)$ Give an efficient algorithm to count the total number of paths in a directed acyclic graph. Analyze your algorithm.

24. $(24.3-2)$ Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. Why doesn't the proof of Theorem 24.6 go through when negative-weight edges are allowed?

25. $(24.3-6)$ We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices

26. $(24.3-9)$ Modify your algorithm from Exercise 24.3-8 to run in $O((V + E) \lg W)$ time. (Hint: How many distinct shortest-path estimates can there be in $V - S$ at any point in time?)

    $(24.3-8)$ Let $G = (V, E)$ be a weighted, directed graph with nonnegative weight function $w : E \to \{0, 1, ..., W\}$ for some nonnegative integer W . Modify Dijkstra's algorithm to compute the shortest paths from a given source vertex s in $O(WV + E)$ time.

27. problem $15-4$ Printing neatly

    Consider the problem of neatly printing a paragraph with a monospaced font (all characters having the same width) on a printer. The input text is a sequence of n words of lengths $l_1, l_2, ..., l_n$, measured in characters. We want to print this paragraph neatly on a number of lines that hold a maximum of M characters each. Our criterion of "neatness" is as follows. If a given line contains words i through j , where $i \leq j$, and we leave exactly one space between words, the number of extra space characters at the end of the line is $M - j + i - \Sigma_{k=i}^{j} l_k$, which must be nonnegative so that the words fit on the line. We wish to minimize the sum, over all lines except the last, of the cubes of the numbers of extra space characters at the ends of lines. Give a dynamic-programming algorithm to print a paragraph of n words neatly on a printer. Analyze the running time and space requirements of your algorithm

موفّق باشید.