DA5300: Data Structures for Data Science(Worksheet-3)

October 27, 2024

Many problems in this document are from "Introduction to Algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press (2022).

1 Heuristic

1. Give an example of a heuristic which is consistent.

2. Give an example of a heuristic which is admissible but not consistent, such that the tree-search fails.

3. Give an example of a heuristic which is admissible but not consistent, such that the tree-search fails tree-search does not fail despite the inconsistency.

4. Given an example of a heuristic which is inadmissible and the graph-search fails.

5. Given an example of a heuristic which is inadmissible and the graph-search succeeds despite the inadmissibility.

6. Prove that if h_1 and h_2 are admissible heuristics, then shown that $h(n) = \min\{h_1(n), h_2(n)\}$ is also an admissible heuristic.

2 Graph Representations

7. Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute the in-degrees?

8. The transpose of a directed graph $G = (V, E)$ is the graph $G^\top = (V, E^\top)$, where $E^\top = \{(v, u) \in V \times V : (u, v) \in E\}$. That is, $G^\top$ is G with all its edges reversed. Describe efficient algorithms for computing $G^\top$ from G, for both the adjacency-list and adjacency-matrix representations of G.

9. The square of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E)$ such that $(u, v) \in E^2$ if and only if G contains a path with at most two edges between u and v. Describe an efficient algorithm for computing G^2 from G for both the adjacency-list and adjacency matrix representations of G.

10. The incidence matrix of a directed graph $G = (V, E)$ with no self-loops is a $|V| \times |E|$ matrix $B = (b_{ij})$ such that

$$b_{ij} = \begin{cases} -1 & \text{if edge } j \text{ leaves vertex } i \\ 1 & \text{if edge } j \text{ enters vertex } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Describe what the entries of the matrix product $BB^\top$ represent, where $B^\top$ is the transpose of B.

3 Topological Sort

For the clothing example in the book, write down the discovery and finish times of the vertices for the following vertex order

11. {shoes, watch , tie, pants, belt, jacket, shirt, socks, undershorts}

12. {jacket, belt, socks, watch , tie, shirt, pants, undershorts shoes, }

13. Another way to topologically sort a directed acyclic graph is to repeatedly find a vertex of in-degree , output it, and remove it and all of its outgoing edges from the graph. Sort the clothing example using this algorithm.

4 Minimum Spanning Tree

14. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on G. 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. Construct an example such that (u, v) is a safe edge for A and is not a light edge of the cut.

15. Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Kruskal's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W?

16. Suppose that all edge weights in a graph are integers in the range from 1 to V. How fast can you make Prim's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W?

17. Professor Borden proposes a new divide-and-conquer algorithm for computing minimum spanning trees, which goes as follows. Given a graph $G = (V, E)$, partition the set of vertices into two sets V_1 and V_2 such that $|V_1|$ and $|V_2|$ differ by at most 1. Let E_1 be the set of edges that are incident only on vertices in V_1, and let E_2 be the set of edges that are incident only on vertices in V_2. Recursively solve a minimum-spanning-tree problem on each of the two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge in that crosses the cut (V_1, V_2) , and use this edge to unite the resulting two minimum spanning trees into a single spanning tree. Either argue that the algorithm correctly computes a minimum spanning tree of G, or provide an example for which the algorithm fails.

5 Single Source Shortest Path

18. Run the Bellman-Ford algorithm on the directed graph of Figure 22.4, using vertex z as the source. In each pass, relax edges in the same order as in the figure, and show the d and π values after each pass. Now, change the weight of edge (z, x) to 4 and run the algorithm again, using s as the source.

19. Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all vertices $v \in V$ of the minimum number of edges in a shortest path from the source s to v. (Here, the shortest path is by weight, not the number of edges.) Suggest a simple change to the Bellman-Ford algorithm that allows it to terminate in $m + 1$ passes, even if m is not known in advance.

20. Let $G = (V, E)$ be a weighted, directed graph with weight function $w \colon E \to \mathbb{R}$. Give an $O(VE)$-time algorithm to find, for all vertices $v \in V$, the value $\delta^*(v) = \min\{\delta(u, v) : u \in V\}$. . 21. Show the result of running DAG-SHORTEST-PATHS on the directed acyclic graph of Figure 22.5, using vertex as the source.

22. Suppose that you change line 3 of DAG-SHORTEST-PATHS to read
3 for the first vertices, taken in topologically sorted order
Show that the procedure remains correct.

23. Run Dijkstra's algorithm on the directed graph of Figure 22.2, first using vertex s as the source and then using vertex z as the source. In the style of Figure 22.6, show the and values and the vertices in set after each iteration of the while loop.

24. Give a simple example of a directed graph with negative-weight edges for which Dijkstra9s algorithm produces an incorrect answer.

25. Give a simple example of a directed graph with negative-weight edges for which Dijkstra9s algorithm does not produce an incorrect answer.

26. Consider a directed graph on which each edge $(u, v) \in E$ 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 t0 v. Interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.

27. Suppose that you are given a weighted, directed graph $G = (V, E)$ in which edges that leave the source vertex s may have negative weights, all other edge weights are nonnegative, and there are no negative-weight cycles. Argue that Dijkstra's algorithm correctly finds shortest paths from in this graph.

28. Give two shortest-paths trees for the directed graph of Figure 22.2 on page 609 other than the two shown.

29. Give an example of a weighted, directed graph $G = (V, E)$ with weight function $w \colon E \to \mathbb{R}$ and source vertex s such that G satisfies the following property: For every edge $(u, v) \in E$, there is a shortest-paths tree rooted at s that contains (u, v) and another shortest-paths tree rooted at s that does not contain (u, v).

30. Let be an arbitrary weighted, directed graph with a negative-weight cycle reachable from the source vertex s.Show how to construct an infinite sequence of relaxations of the edges of G such that every relaxation causes a shortest-path estimate to change.

6 All Pair Shortest Path (APSP)

31. Run SLOW-APSP on the weighted, directed graph of Figure 23.2, showing the matrices that result for each iteration of the loop. Then do the same for FASTER- APSP.

32. Run the Floyd-Warshall algorithm on the weighted, directed graph of Figure 23.2. Show the matrix that results for each iteration of the outer loop.

33. Use Johnson's algorithm to find the shortest paths between all pairs of vertices in the graph of Figure 23.2. Show the values of and computed by the algorithm.

34. Professor Greenstreet claims that there is a simpler way to reweight edges than the method used in Johnson's algorithm. Letting $w^* = \min\{w(u,v) : (u,v) \in E\}$, just define $\hat{w}(u,v) = w(u,v) - w^*$ for all the edges $(u,v) \in E$. What is wrong with professor's method of re-weighting?

35. Professor Michener claims that there is no need to create a new source vertex in line 1 of JOHNSON. He suggests using $G' = G$ instead and letting s be any vertex. Give an example of a weighted, directed graph G for which incorporating the professor's idea into JOHNSON causes incorrect answers.

7 Maximum Flow

36. Suppose that, in addition to edge capacities, a flow network has vertex capacities. That is each vertex has a limit on how much flow can pass through . Show how to transform a flow network with vertex capacities into an equivalent flow network without vertex capacities, such that a maximum flow in has the same value as a maximum flow in . How many vertices and edges does have?

37. In Figure 24.1(b), what is the net üow across the cut ? What is the capacity of this cut?

38. In the example of Figure 24.6, what is the minimum cut corresponding to the maximum flow shown? Of the augmenting paths appearing in the example, which one cancels flow?

39. Show how to find a maximum flow in a flow network by a sequence of at most augmenting paths. (Hint: Determine the paths after finding the maximum flow.)