

## Study Questions

## Graph Algorithms

In addition to the homeworks already assigned, and the study questions for the midterm, the following are good study questions for the final exam.

1. Suppose  $G$  is an undirected, connected weighted graph such that  $G$  is not the complete graph but every edge in  $G$  has positive weight. Create a complete graph,  $H$ , having the same vertex set as  $G$ , such that for each pair of vertices,  $v$  and  $u$ , in  $G$ , the edge  $(v, u)$  in  $H$  has weight equal to the shortest-path distance between  $v$  and  $u$  in  $G$ . Show that the edge weights in  $H$  satisfy the triangle inequality.
2. Show that the number of vertices of odd degree in a tree is even.
3. Consider a greedy algorithm for the VERTEX-COVER problem, where we repeatedly choose a vertex with maximum degree, add it to our cover, and then remove it and all its incident edges. Show that this algorithm does not, in general, produce a 2-approximation.
4. (CS 265) Suppose you are working for a cartography company, that is, a company that makes maps. Your job is to design a software package that can take as input the map of some region,  $R$ , and label as many of the cities of  $R$  as possible. Each of the  $n$  cities in such a region,  $R$ , is given by an  $(x, y)$  coordinate for the center of that city. Assume, for the sake of simplifying the problem, that the label,  $L_c$ , for each city,  $c$ , is a rectangle (which will contain the name of the city,  $c$ ) whose lower-right corner is the  $(x, y)$ -location for  $c$ . The labels for two cities,  $c$  and  $d$ , *conflict* if  $L_c$  intersects  $L_d$ . Given your extensive algorithmic background, you realize that you can model this problem with a graph,  $G$ , where you create a vertex in  $G$  for each city and connect cities  $c$  and  $d$  with an edge if their labels conflict. Let  $d = 2m/n$  be the average degree of the vertices in  $G$ , where  $m$  is the number of edges in  $G$ . Describe an  $O(d)$ -approximation algorithm for finding the largest number of mutually nonconflicting labels for the cities in a given region  $R$ .
5. (CS 265) Suppose you are preparing an algorithm for the problem of optimally drilling the holes in an aluminum plug plate to allow it to do a spectrographic analysis of a set of galaxies. Based on your analysis of the robot drill device, you notice that the various amounts of time it takes to move between drilling holes satisfies the triangle inequality. Nevertheless, your supervisor does not want you to use the MST approximation algorithm or the Christofides approximation algorithm. Instead, your supervisor wants you to use a nearest-neighbor greedy algorithm for solving this instance of METRIC-TSP. In this greedy algorithm, one starts with city number 1 as the “current” city, and then repeatedly chooses the next city to add to the tour to be the one that is closest to the current city (then making the added city to be the “current” one). Show that your supervisor’s nearest-neighbor greedy algorithm does not, in general, result in a 2-approximation algorithm for METRIC-TSP.
6. Draw a flow network with 9 vertices and 12 edges. Illustrate an execution of the Ford-Fulkerson algorithm on it, showing each augmenting path and the change in flow with each augmentation. Show the min-cut in this network as well, using dashed lines.
7. Illustrate the execution of the Edmonds-Karp algorithm in your flow network from the previous exercise.
8. Show that, given a maximum flow in a network with  $m$  edges, a minimum cut of  $N$  can be computed in  $O(m)$  time.

9. Draw a bipartite graph with 5 vertices on each side, 10 edges total, and show how to use network flow to find a maximum matching in this graph.
10. What is the worst-case running time of the Ford-Fulkerson algorithm if all edge capacities are bounded by a constant,  $c$ ?
11. Give an algorithm that determines, in  $O(n + m)$  time, whether a graph with  $n$  vertices and  $m$  edges is bipartite.
12. In the context of the baseball elimination problem, one can show that if  $w_i + g_i \leq w_k + g_k$  and team  $k$  is eliminated, then team  $i$  is also eliminated. Use this fact to show that among a set of  $n$  teams, one can determine all the eliminated teams by solving  $O(\log n)$  maximum flow problems.
13. Recall that a *vertex cover* for a graph,  $G$ , is a set of vertices,  $C$ , such that every edge in  $G$  is incident to one of the vertices in  $C$ . The problem of finding a smallest vertex cover is useful in network monitoring and other applications, but it is a difficult problem for general graphs. Show that the problem of finding the size of a smallest vertex cover in a bipartite graph can be solved in polynomial time.
14. Imagine that you are working on creating a flow for a set of packets in a media stream, as described in the introduction to this chapter. So you are given a network,  $G$ , with a source,  $s$ , and sink,  $t$ , together with bandwidth constraints on each edge, which indicate the maximum speed that the communication link represented by that edge can support. As mentioned before, your goal is to produce a maximum flow from  $s$  to  $t$ , respecting the bandwidth constraints on the edges. Suppose now, however, that you also have a bandwidth constraint on each router in the network, which specifies the maximum amount of information, in bits per second, that can pass through that node. Describe an efficient algorithm for finding a maximum flow in the network,  $G$ , that satisfies the bandwidth capacity constraints on the edges as well as the vertices. What is the running time of your algorithm?
15. Suppose, as an interview question, you are told that you have a goat and a wolf that need to go from a node,  $s$ , to a node,  $t$ , in a directed acyclic graph,  $G$ . To avoid the wolf eating the goat, their paths must never share an edge. Describe a polynomial-time algorithm for finding two edge-disjoint paths in  $G$ , if such paths exist, to provide a way for the goat and the wolf to go from  $s$  to  $t$  without risk to the goat.
16. Suppose a friend of yours has created a simulation game based on J.R.R. Tolkien's epic *The Lord of the Rings*. The game environment is Middle Earth, which is populated by various noble creatures, including hobbits, humans, dwarves, and elves. Unfortunately, these noble creatures are under attack and need to get to safe havens, known as "strongholds." Some strongholds are larger than others, of course, and each stronghold,  $s$ , can only hold some number,  $N_s$ , of these creatures. Initially, let us assume each stronghold is empty, and the noble creatures are living in various regions, with each region,  $r$ , containing some number,  $N_r$ , of noble creatures. Moreover, we know, for each region,  $r$ , the set of strongholds,  $S_r$ , that can be reached from  $r$  in at most three days' travel. Your job is to figure out how to move the maximum number of noble creatures possible from the regions where they currently live to the various strongholds in three days' time while not overcrowding any stronghold. Describe and analyze an efficient algorithm to solve this game.
17. Provide an example of an instance of the stable marriage algorithm, with 5 men and 5 women, and show how each man gets his top choice, but no woman gets her top choice.

18. In a famous experiment, Stanley Milgram told a group of people in Kansas and Nebraska to each send a postcard to a lawyer in Boston, but they had to do it by forwarding it to someone that they knew, who had to forward it to someone that they knew, and so on. Most of the postcards that were successfully forwarded made it in 6 hops, which gave rise to the saying that everyone in America is separated by “six degrees of separation.” The idea behind this experiment is also behind a technique, called *probabilistic packet marking*, for doing traceback during a distributed denial-of-service attack, where a website is bombarded by connection requests. In implementing the probabilistic packet marking strategy, a router,  $R$ , will, with some probability,  $p \leq 1/2$ , replace some seldom-used parts of a packet it is processing with the IP address for  $R$ , to enable tracing back the attack to the sender. It is as if, in the Milgram experiment, there is just one sender, who is mailing multiple postcards, and each person forwarding a postcard would, with probability,  $p$ , erase the return address and replace it with his own. Suppose that an attacker is sending a large number of packets in a denial-of-service attack to some recipient, and every one of the  $d$  routers in the path from the sender to the recipient is performing probabilistic packet marking.
  - (a) What is the probability that the router farthest from the recipient will mark a packet and this mark will survive all the way to the recipient?
  - (b) Derive a good upper bound on the expected number of packets that the recipient needs to collect to identify all the routers along the path from the sender to the recipient.
19. What is the degeneracy of a tree?
20. What is the degeneracy of a cycle?
21. What is the degeneracy of the following graph:
 

<https://i.stack.imgur.com/MysuI.png>
22. What is the arboricity of a graph?
23. What is the definition of a power law? Give an example of a graph with a power law?
24. What is the BA preferential attachment model?
25. Describe an efficient algorithm for computing the local clustering coefficient for each vertex in a graph of  $n$  vertices,  $m$  edges, and degeneracy  $d$ .
26. (CS 163 students only) Prove the 6-color theorem for planar graphs.
27. (CS 265 students only) Prove the 5-color theorem for planar graphs.