CS 577: Introduction to Algorithms — Network Flow Review Problems

Network Flow

In the following questions, a flow network is a described by a 4-tuple, (G, c, s, t), where $G = (V, E)$ is a directed acyclic graph, c_e is the non-negative integer capacity on edges $e \in E$, s is the source node, and t is the sink node.

1. Given a flow network, describe an algorithm to determine whether a flow f is a maximum flow.

2. Let (S, T) and (S', T') be two minimum cuts of a flow network, prove that $(S \cap S', T \cup T')$ and $(S \cup S', T \cap T')$ are also minimum cuts.

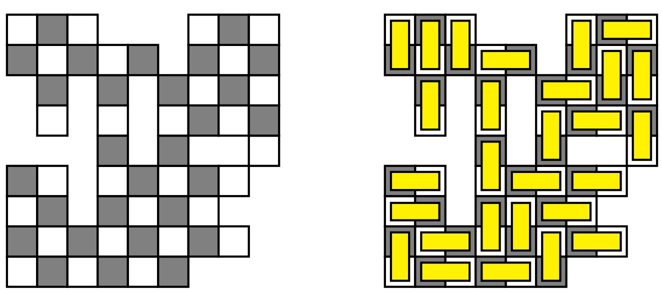
3. (Week 10 Discussion Problem $1\,(b)$) Define a best minimum cut to be any minimum cut with the smallest number of edges. Describe an efficient algorithm to determine whether a given flow network contains a unique best minimum cut.

4. An edge in a flow network is called lower-binding if reducing its capacity by one unit decreases the maximum flow in the network. Describe and analyze an $O\left(|V||E|\right)$ algorithm for finding all the lower-binding edges given a flow network as well as a maximum flow $f^\star$.

Applications of Network Flow

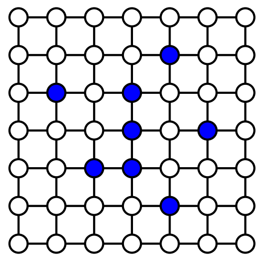
5. Suppose you work for Facebook and you want to find groups of users which are "close-knit". Informally, you have a parameter α in mind, and you want to find a group of users S such that the number of friends among the users in S, call this $f\left(S\right)$, is at least an α fraction of the total number of users in the set. Formally, you are given a graph $G = (V, E)$ and a parameter α between 0 and 1. For any $S \subseteq V$, define $f\left(S\right)$ to be the number of edges in E with both endpoints in S. Give an efficient algorithm which determines whether there exists a set $S \in V$ such that $\dfrac{f\left(S\right)}{|S|} \geq \alpha$.

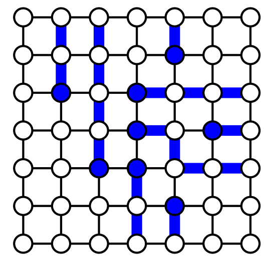
6. Suppose you are given an $n \times n$ checkerboard with some of the squares deleted. You have a large set of dominoes, just the right size to cover two squares of the checkerboard. Describe and analyze an algorithm to determine whether one can tile the board with dominoes: each domino must cover exactly two squares that are not deleted, and each square that is not deleted must be covered by exactly one domino. For example, if $n = 9$ and the squares are deleted as in the following diagram on the left, the algorithm should return the following yellow tiles in the diagram on the right.



7. An $n \times n$ grid is an undirected graph with n^2 vertices organized into n rows and n columns. Denote the vertex in row i and column j by (i, j). Every vertex in the grid has exactly four neighbors, except for the boundary vertices, which are the vertices (i, j) such that $i = 1, i = n, j = 1$, or $j = n$. Let $\left(x_1, y_1\right), \left(x_2, y_2\right), ..., \left(x_m, y_m\right)$ be distinct vertices, called terminals, in the $n \times n$ grid. The escape problem is to determine whether there are m vertex-disjoint paths in the grid that connect the terminals to any m distinct boundary vertices. Describe and

analyze an efficient algorithm to solve the escape problem. For example, if $n = 7$ and terminals labelled as blue nodes in the diagram on the left, the algorithm should return the blue paths in the diagram on the right.





8. Given an $m \times n$ array of non-negative real numbers, we want to round A to an integer matrix, by replacing each entry x in A with either $\lfloor x \rfloor$ or $\lceil x \rceil$, without changing the sum of entries in any row or column of A. Describe and analyze an efficient algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible. For example, if A is the matrix on the left, the algorithm should return the matrix on the right,

$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}.$$