Homework 12

2019.05.23

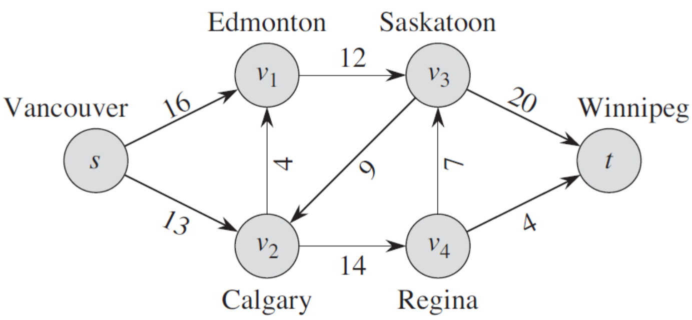
Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. Exercises 26.1-7

Suppose that, in addition to edge capacities, a flow network has **vertex capacities**. That is each vertex v has a limit $l(v)$ on how much flow can pass through v. Show how to transform a flow network $G = (V, E)$ with vertex capacities into an equivalent flow network $G' = (V', E')$ without vertex capacities, such that a maximum flow in G' has the same value as a maximum flow in G. How many vertices and edges does G' have?

2. Exercises 26.2-3

Show the execution of the Edmonds-Karp algorithm on the flow network of Figure 26.1(a).



(a)

3. Exercises 26.2-11

The edge connectivity of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how to determined the edge connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges. (Find the value of edge connectivity.)

4. Exercises 26.2-13

Suppose that you wish to find, among all minimum cuts in a flow network G with integral capacities, one that contains the smallest number of edges. Show how to modify the capacities of G to create a new flow network G' in which any minimum cut in G' is a minimum cut with the smallest number of edges in G.

5.

The vertex connectivity of an undirected graph is the minimum number k of vertices that must be removed to disconnect the graph. When we remove a vertex, we must also remove the edges incident to it. For example, the vertex connectivity of a tree is 1, and the vertex connectivity of a cyclic chain of vertices is 2. Show how to determine the vertex connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges. (If the graph is complete graph, the vertex connectivity is $|V| - 1$.)

6. Exercise 26.3-4

A ***perfect matching*** is a matching in which every vertex is matched. Let $G = (V, E)$ be an undirected bipartite graph with vertex partition $V = L \cup R$, where $|L| = |R|$. For any $X \subseteq V$, define the ***neighborhood*** of X as

$$N(X) = \{ y \in V : (x, y) \in E \text{ for some } x \in X \}$$

that is, the set of vertices adjacent to some member of X. Prove ***Hall's theorem***: there exists a perfect matching in G if and only if $|A| \leq |N(A)|$ for every subset $A \subseteq L$.

7. EXT 10-1

There are two extended ways used to find the augmenting path that we have mentioned in class (refers to slides p.14, Unit 10), please design an efficient algorithm with the argument of second method to find the augmenting path. Argue that your algorithm is correct and also analyze the time-complexity.