Network Flow

Tom Morgan

December 8, 2006

What it is

Say you have a network of pipes. At one point there is a source of water and at the end there is a sink. Each of the pipes has some capacity for water per unit time. You want to find the maximum amount of water that flows from the source to the sink per unit time. This is the problem maximum flow, which can be found with a network flow algorithm. We will focus on the Ford-Faulkerson algorithm because it is simple and efficient.

The Algorithm

The algorithm exists in three steps.

In the first step, all we have to do is find a path from the source to the sink. For this we can use any kind of path finding algorithm. Depth First Search is the simplest, Dikstra's often yields efficient results but Breadth First Search is the only method that has result in a polynomial bound on the general case.

Now that we have a path from the source to the sink, we add the maximum amount of flow that can be pushed through the pipes: the minimum of the capacities of the pipes involved. However, what if the path we used is a bad one? We need to take action to correct any potential mistakes. The way we do this is by pushing flow back along the path from the sink to the source. We subtract the amount we pushed through from the capacity in the direction we followed originally, and add it in the reverse direction (the graph is directed.)

Finally, to find all the flow, we simply keep performing the first two steps until there is no path from the source to the sink.

Minimum Cut

So far, we have been talking about Network Flow solely in terms of maximum flow. However, Network Flow can also, without any modifications, be used to find the minimum cut. The minimum cut is defined as the minimum cost it takes to eliminate any paths between to vertices. In this context, rather than each edge's (or pipe's) weight being its capacity, it is the cost it takes to remove that edge. If you need to know which edges you are removing, they are simply the first edge whose cost is reduced to zero by the second step in the algorithm.

Bipartite Matching

The most common form of Network Flow that you will find on a contest is what is known as Bipartite Matching. Bipartite Matching is, wonderfully enough, the absolute simplest form of Network Flow. In bipartite matching, there is a source, a sink and two groups of vertices. Each group is arranged in a column (were one to describe them visually). The source has an edge to each vertex in the first column. Each vertex in the second column has an edge to the sink. The vertices in the first column have some edges connecting them to some edges in the second column. All of the edge weights are initially one.

Bipartite Matching is a time when we are free to use a Depth First Search without fear. This is because for Bipartite Matching it yields a run time of $O(E * V)$.

A classic example of this Bipartite Matching (which you will find in the USACO training pages) is the problem of matching cows to stalls. Say you have N cows and M stalls and each cow is only willing to stay in a specific subset of the stalls. Find the maximum number of cows that can be placed in stalls. In this case, the first column would be the cows, second column would be the stalls and the connections between them would be edges representative of which stalls a cow can stay in (if a cow 1 is willing to stay in stall 3 there is an edge from cow 1 to stall 3.)

Problems

Shamlessly stolen from a past SCT lecture.

1. Code up network flow; this is often the hardest part of a problem. It is a fairly long one, with plenty of places to make mistakes. The more you do it, the faster you will get at it.

2. Farmer John is giving his cows gifts in an attempt to improve moral. He has a set of gifts to give out, and knows how many gifts he can give each cow (different for each cow) before more gifts cease to have an effect. Help him distribute the gifts among the cows to maximize the total happiness of his herd.

3. You are working for quality control for a milk distribution company, and you just learned a shipment of bad milk was sent out. You know where it's going, but not how it's going to get there. The distribution system consists of several warehouses, with trucks running between warehouses. You want to prevent the milk from being delivered by shutting down certain trucks; each truck will result in a certain amount of lost money. Find the trucks that should be stopped to minimize lost money while ensuring that the milk is not delivered.

4. The cows are setting up umbrellas on the beach for shade. The beach can be represented as a 50 50 grid, with cows sitting in certain squares, and an umbrella covering two orthogonally adjacent squares. The cows want to shade as many of themselves as possible, but are firmly against waste and do not want to shade any squares that do not have cows. Find out the maximum number of cows that can be covered without covering any cow-less area.

5. Farmer John has a number of cows capable of milking themselves. Whenever a cow enters the barn, she instantaneously hooks herself up to the milking machine and produces milk at a constant rate (potentially different for each cow) for the duration of her stay in the barn, and unhooks herself immediately before leaving the barn. Farmer John can only measure the aggregate output of all cows in the barn at any given time; given a set of these measurements and the times when each cow entered and left the barn, find the milk production rate of each cow.