CS 577: Introduction to Algorithms 11/5/2015

Homework 8

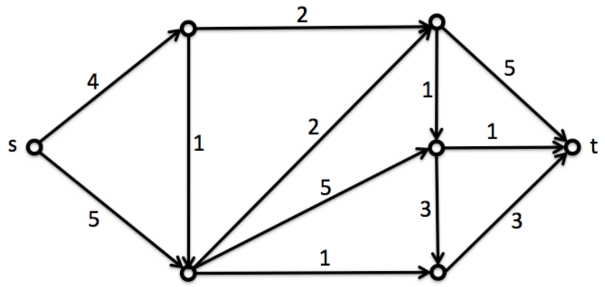
Instructor: Dieter van Melkebeek TA: Kevin Kowalski, Andrew Morgan, Bryce Sandlund

This assignment covers network flow. Good luck!

Review problems

1. Consider a network with integer capacities. An edge is called *upper-binding* if increasing its capacity by one unit increases the maximum flow value in the network. An edge is called *lower-binding* if reducing its capacity by one unit decreases the maximum flow value in the network.

   (a) For the network G below determine a maximum flow f^*, the residual network G_{f^*}, and a minimum cut. Also identify all of the upper-binding edges and all of the lower-binding edges.



   (b) Develop an algorithm for finding all the upper-binding edges in a network G when given G and a maximum flow f^* in G. Your algorithm should run in linear time.

   (c) Develop an algorithm for finding all the lower-binding edges in a network G when given G and an integer maximum flow f^* in G. Your algorithm should run in linear time.

2. A *vertex cover* of a graph $G = (V, E)$ is a collection of vertices $C \subseteq V$ such that every edge $e \in E$ has at least one vertex in C.

   Show that for bipartite graphs, the minimum size of a vertex cover equals the maximum size of a matching.

Graded written problem

3. [10 points] Let $G = (L \cup R, E)$ be a bipartite graph with $L \cap R = \emptyset$ and $E \subseteq L \times R$. Suppose you are given a matching M_L covering $L' \subseteq L$ and a matching M_R covering $R' \subseteq R$. Show how to efficiently construct a matching M covering $L' \cup R'$.

Additional written problem

4. A given network can have many minimum *st*-cuts.

   (a) Determine precisely how large the number of minimum *st*-cuts in a graph can be as a function of n.

   (b) Show that if (S_1, T_1) and (S_2, T_2) are both minimum *st*-cuts in a given network, then so is $(S_1 \cup S_2, T_1 \cap T_2)$. How does this generalize to more than 2 *st*-cuts?

   (c) Construct a polynomial-time algorithm that, given a network, generates a collection of minimum *st*-cuts $(S_1, T_1), (S_2, T_2), \ldots$ such that every minimum cut of the network can be written as

   $$(\cup_{i \in I} S_i, \cap_{i \in I} T_i)$$

   for some subset I of indices.

Optional programming problem

5. [2.5 points] Solve SPOJ problem Potholers (problem code POTHOLE).

Challenge problem

6. Problem J from the 2007 ACM-ICPC World Finals (see next page). Your algorithm should run in time polynomial in $n \doteq R + T$.

Problem J

Tunnels

Input File: tunnels.in

Curses! A handsome spy has somehow escaped from your elaborate deathtrap, overpowered your guards, and stolen your secret world domination plans. Now he is running loose in your volcano base, risking your entire evil operation. He must be stopped before he escapes!

Fortunately, you are watching the spy's progress from your secret control room, and you have planned for just such an eventuality. Your base consists of a complicated network of rooms connected by non-intersecting tunnels. Every room has a closed-circuit camera in it (allowing you to track the spy wherever he goes), and every tunnel has a small explosive charge in it, powerful enough to permanently collapse it. The spy is too quick to be caught in a collapse, so you'll have to strategically collapse tunnels to prevent him from traveling from his initial room to the outside of your base.

Damage to your base will be expensive to repair, so you'd like to ruin as few tunnels as possible. Find a strategy that minimizes the number of tunnels you'll need to collapse, no matter how clever the spy is. To be safe, you'll have to assume that the spy knows all about your tunnel system. Your main advantage is the fact that you can collapse tunnels whenever you like, based on your observations as the spy moves through the tunnels.

Input

The input consists of several test cases. Each test case begins with a line containing integers R ($1 \leq R \leq 50$) and T ($1 \leq T \leq 1000$), which are the number of rooms and tunnels in your base respectively. Rooms are numbered from 1 to R. T lines follow, each with two integers x, y ($0 \leq x,y \leq R$), which are the room numbers on either end of a tunnel; a 0 indicates that the tunnel connects to the outside. More than one tunnel may connect a pair of rooms.

The spy always starts out in room 1. Input is terminated by a line containing two zeros.

Output

For each test case, print a line containing the test case number (beginning with 1) followed by the minimum number of tunnels that must be collapsed, in the worst case. Use the sample output format and print a blank line after each test case.

Sample Input

```
4 6
1 2
1 3
2 4
3 4
4 0
4 0
4 6
1 2
1 3
1 4
2 0
3 0
4 0
0 0
```

Output for the Sample Input

```
Case 1: 2

Case 2: 2
```