CMSC 401: Algorithm Analysis with Advanced Data structures
Spring 2017
Class Project

Instructor: Thang Dinh
Virginia Commonwealth University
School of Engineering

Due: Thu, May 06, 2017

Given an undirected graph $G = (V, E)$, for any subset of nodes $S \subseteq V$ we can construct a graph G_S from G by removing all nodes in S together with their incident edges. In the critical node problem (CNP), we are given an integer $1 \leq k \leq |V|$ and need to find a subset S of size k such that the graph G_S has the minimum pair-wise connectivity. Here pairwise connectivity of a graph is defined as the number of pairs of connected vertices in the graph.

Input: The file "cnp.in" includes multiples lines. The first line contains three integers $1 \leq n \leq 1000, 1 \leq m \leq 100000$ and $1 \leq k \leq n$ that correspond to the number of nodes, edges, and the size of S. Each of the following m lines contain two integers u and v, separated by one space, to denote an edge from u to v. Nodes are numbered from 1 to n.

Output: The file "cnp.out" contains exactly 2 lines. The first line contains an integer P that is the minimum pairwise connectivity of G_S. The second line contains exactly k integers which are the id of the nodes in S.

Sample input/output:

cnp.in	cnp.out
7 11 2	4
1 2	3 4
1 3	
1 4	
2 4	
3 4	
3 5	
3 6	
4 6	
5 6	
5 7	
6 7	

Explain of the output: After removing two nodes 3 and 4 and their incident edges from the graph, we obtain a graph G_S with two connected components $C_1 = \{1, 2\}$ and $C_2 = \{5, 6, 7\}$. The number of connected pairs in the component C_1 and C_2 are one and three, respectively. Thus the total number of connected pairs (pairwise-connectivity) is 4.

Submission: Your submission in Blackboard should include the following items

I. Your program in Java/C++ that solves the above problem following the above input/output format. A makefile and/or compiling instruction should be included if you have multiple source files. Your program should not take more than 5 minutes to terminate on any graph within the limits described in the Input section.

II. A report outline the results of your program on random graphs and different k values. The report should have at least two parts. In the first part, you fix $k = 5$ and run your program for graphs of sizes 50, 100,..., 500. In the second parts, you run your program on a random graph of size 100 and output the pairwise connectivity in G_S when $k = 5, 10, ..., 50$. For each run, output the number of nodes, edges, the number k, and the pairwise connectivity of G_S.

To generate random graphs, you can write your own program or use the source code here: `http://algs4.cs.princeton.edu/41graph/GraphGenerator.java.html`. Other options include Boost C++ library `http://www.boost.org/doc/libs/1_54_0/libs/graph/doc/erdos_renyi_generator.html` or Igraph `http://igraph.org/python/doc/tutorial/tutorial.html`

Grading: The grading will be based on your report and your (relative) performance of your program on unpublished test files.