SNU ECE. Advanced Programming Methodology **Due: 12:30pm Nov 21 2017**

Homework 3

Lecturer: Kyomin Jung

1. Karger's Algorithm (15p)

State the Karger's algorithm for the minimum cut problem for undirected edge-weighted graph G, and prove that it computes a minimum cut of G with probability 1-o(1).

2. Max-flow Min-cut Algorithm (Coding Problem, 35p)

A. (10p) An undirected weighted flow network G is given as a 200×200 adjacency matrix. Vertex number is given from 1 to 200. The capacities of the graph represented by the adjacency matrix are same as weights of the graph at HW2 problem 4. Read the data file and make a program that implements the graph structure as a linked list. You are allowd to reuse your linked list implementation at HW2 problem 4. Let the vertex 1 be the source vertex s and vertex 150 be the target vertex t. Make a program that implements the BFS (breath first search) to find a shortest s-t path (when the edge weights are considered to be all 1). Output the computed shortest path and its length. The length means the number of edges in the shortest path. Your output should be something like:

Length: xx

(1,x),(x,x),…,(x,150)

B. (15p) Make a program that implements the Edmond-Karf algorithm using the linked list. The source and the target vertices are the same as 2-A. If you want, you are allowed to use additional array structures. Output the computed maximum flow from the source vertex to the target vertex. In the flow vector addition step of the Edmond-Karf algorithm, for example, if the flow of one directed edge is 2, and the flow of the other direction is 5, these values should be updated to be 0 and 3, respectively. I.e., after the flow vector addition, for any undirecgted-edge, only at most one direction should have a non-zero flow value. Your output should include the flow of each edge connected to the source vertex s. Your output should be something like:

Total:xx.xx

(1,x)Flow:xx.xx

(1,x)Flow:xx.xx

…

(1,x)Flow:xx.xx

C. (10p) Make a program that finds a minimum (S,T)-cut so that s belongs to S and t belongs to T using your results in 2-A and 2-B. The source and target vertex are the same as 2-B. If you want, you are allowed to use additional array structures. Output the total weight of the minimum cut and the cut edge list. Your output should be something like:

    Total:xx.xx
    (x,x),(x,x), …,(x,x)

(a) Data: Get the data file from eTL Class Announcements board (graph.txt).
(b) Language: Use one of C/C++/Java/Python/Matlab.
(c) README file: You should provide a README file which explains the programming environments (OS, compiler), and instructions for compiling and running your program.

Deliverables & how to submit : Submit 1 zipped file of your source codes, a README file, and a doc file containing the output results. Submit to eTL. You don't need to submit hardcopy version of this problem.

3. P vs NP (20p)

A. State the two definitions of NP, and the definition of NP-complete. (8p)

B. Prove that the Eulerian path problem is in P, and the Hamiltonian path problem is in NP. (12p)

4. NP-Completeness (30p)

A. Prove that the Clique problem is NP-complete by assuming that the 3SAT problem is NP-complete. (12p)

B. The Vertex Cover (VC) problem is as follows. For an undirected graph G = (V, E), a subset S of V is a vertex cover if every edge in E has at least one end vertex in S. For a given graph G and an integer k, the VC problem is to decide whether G has a vertex cover of size at most k. Prove that the VC problem NP-complete. (Hint: a reduction from the Independent Set problem.) (8p)

C. The maximum-2-satisfiability problem (MAX-2-SAT) is as follows. The input is a 2-CNF formula (i.e. two literals per clause), and an integer k. The task is to determine whether there is a Boolean assignment that satisfies at least k many clauses. MAX-2-SAT is known to be NP-complete. (by a reduction from 3SAT. Think about it if you want.) Now, consider the Max-Cut problem on an undirected unweighted graph. I.e., for a given graph G and an integer k, the problem is to decide whether there is a cut of size at least k or not. Prove that the Max-Cut problem is NP-complete by assuming that MAX-2SAT is NP-complete. (10p)

5. Bonus Problem (Don't submit your answer)

Prove that the 2-SAT problem is in P. Hint: consider the following graph construction. For a given 2-CNF formula F on n Boolean variables, consider a directed graph G on 2n nodes as follows. The set of all the 2n literals of F form the nodes of G (x and $\neg x$ become two different nodes). For each 2-clause ($x \vee y$) of F, where x and y are literals, draw two directed edges of G : $\neg x \rightarrow y$, and $\neg y \rightarrow x$. Think of some relevant problem on G.

- Submit your answer sheets (except for Problem 2) either in the class on Nov 21st , or directly to TA. You should submit the answer file of Problem 2 to eTL.
- Write your answers by your handwriting in English or Korean. No computer typed answers will be accepted.
- Use separate sheet of paper for each problem.
- No delay will be allowed. If you submit your answers later than 12:30pm on Nov 21st , your score for the delayed parts will be 0.