CS 584/684 Spring 2017 Homework 8 – due noon, Wednesday, May 31 2017

Your solutions to problems 1-4 should be type-set in LaTeX and submitted in both `.tex` and `.pdf` form, with file names `hw8.tex` and `hw8.pdf`. These two files, plus any additional source files invoked from your `.tex` file (such as pictures), should be bundled together into a single `.zip` file named *your-last-name*`-tex-hw8.zip`. Your code for problem 5 should be submitted in a single, separate `.zip` file named *your-last-name*`-code-hw8.zip` with contents as described below in the description of problem 5.

Submit by emailing to `hamialex@pdx.edu` including the zip file as a separate attachment and including "`CS584 HW8`" in the subject line.

All algorithms must be accompanied by proofs of correctness and of running time.

1. [10 pts] Fill in some details of the max-flow min-cut theorem given in the lecture notes:

(a) Prove formulae (1), (2), and (3) on p. 3.

(b) In step (2) $\Rightarrow$ (3) of the proof on p. 6, prove that f' obeys skew symmetry.

2. [10 pts] Do CLRS Problem 26-1.

3. [10 pts] Do CLRS 17.3-6. Perform the amortized analysis twice, first using the accounting method and then using the potential method.

4. [10 pts] Do CLRS Problem 17-2 parts (a) and (b).

5. [10 pts] Implement the Edmonds-Karp max flow algorithm (i.e. the "short pipe" instantiation of the Ford-Fulkerson method).

For this problem, you should implement all data structures yourself. Please try to avoid the temptation to Google solutions. If you cannot help yourself, then, as always, give appropriate credit.

Your program should take one command line argument, which is the name of an input file. The format of that file will be as follows:

- First line contains a number C of test cases in the file, where $0 \leq C \leq 10^6$.
- Then come C test cases, each describing a flow network, and consisting of:
  - A single line containing the number of vertices V in the network, where $1 \leq V \leq 10^6$.
  - A single line containing two space-separated numbers s and t, where s is the vertex number of the source vertex and t is the vertex number of the sink vertex, and $1 \leq s, t \leq V$.
  - A single line containing the number of edges E in the network, where $0 \leq E \leq 10^9$.
  - For each edge $i : 1 \leq i \leq E$, a single line describing edge e_i by three space-separated numbers v_1 v_2 c, where v_1 and v_2 are the vertex numbers of the edge's endpoints, with $1 \leq v_1, v_2 \leq V$, $v_1 \neq v_2$, and c is an integer capacity, with $1 \leq c \leq 10^6$.

  You may assume that if (v_1, v_2) is among the edges, then (v_2, v_1) is not.

Your program should output (to `stdout`) one line for each test case, of the form "`Case i:` n" where n is the value of a maximum flow (or the capacity of a minimum cut) of the network for case i.

Example Input:

```
2
2
2 1
1
2 1 10
6
1 6
9
1 2 3
2 3 2
3 6 3
4 2 3
3 4 1
5 3 3
1 4 2
4 5 3
5 6 2
```

Corresponding Example Output:

```
Case 1: 10
Case 2: 4
```

Warning: If your output format is not correct (even spacing), you will get no credit; this problem will be graded by doing a `diff` against a standard output file.

Place your code (one or more source files) together with a `Makefile` in a fresh subdirectory with the name *your-last-name*`-code-hw8`. Then create a single `zip` archive with the name *your-last-name*`-code-hw8.zip` containing just that directory and its contents. It should be possible to build an executable file called `hw8` and test it on an input file `/path/to/foo` by the following steps:

1. `unzip` *your-last-name*`-code-hw8.zip`
2. `cd` *your-last-name*`-code-hw8`
3. `make`
4. `./hw8 /path/to/foo`