BLG 336E - Analysis of Algorithms II
2019/2020 Spring
Final Project

- You should write all your code in C++ language. Your code should be run with the command line arguments specified for each question.
- Your code should be able to be compiled with default g++ compiler and run under Ubuntu OS. **Even if you are writing your code on a different OS, you should check it via ITU SSH.**
- This is a Final Course Project Assignment, cheating is absolutely unethical and morally unacceptable. It will be punished by a negative grade. Also disciplinary actions will be taken.
- For every part of the homework, programs should be run with different command line arguments. **The codes not using these arguments or giving output in a different layout will not be graded**.

1 Network Flow (25 pts)

Implementation [20 pts]
In this part, you are going to implement Push-Relabel algorithm [1] to calculate maximum flow in a network. You are required to complete the class definitions, in **q2.cpp**, by following the pseudocodes in Figures 1, 2, 3, 4. You may want to read the chapter 26.4 from CLRS [2]. An example output for **test.txt** is given below. First line is number of nodes in the graph and other lines are the edges of the graph in **test.txt**.

```
g++ q2.cpp -o q2
./q2 test.txt
Maximum Flow: 20
```

Report [5 pts]
Compare Ford Fulkerson and Push Relabel algorithms in your **own** words. [Max 10 sentences]

```
PUSH(u,v):
// e(u): Excess flow a u
// cf(u,v): Residual capacity, c(u,v) - f(u,v)
delta(u,v) = min(e(u), cf(u,v))
f(u,v) = f(u,v) + delta(u,v) // Increase the flow on edge (u,v)
f(v,u) = f(v,u) - delta(u,v) // Decrease the flow on edge (v,u)
e(u) = e(u) - delta(u, v) // Update the excess flow
e(v) = e(v) + delta(u,v) // Update the excess flow
```

Figure 1: Push

```
RELABEL(u):
// h(u): height of u
// cf(u,v): Residual capacity, c(u,v) - f(u,v)
h(u) = min(h(v) for all v such that cf(u,v) > 0) + 1
```

Figure 2: Relabel

```
INITIALIZE PREFLOW(G, s):
for each vertex v
    h(v) = 0
    e(v) = 0
for each edge (u,v)
    f(u,v) = 0
h(s) = |V| // height of source: number of vertices
for each vertex v adjacent to source
    f(s,v) = c(s,v) // Initialize with capacity
    e(v) = c(s,v)
    e(s) = e(s) - c(s,v)
```

Figure 3: Initialize Preflow

```
GENERIC PUSH RELABEL(G):
INITIALIZE PREFLOW(G, s):
while there exists an applicable push or relabel operation // Overflowing vertices
    select and applicable push or relabel operation and perform it
```

Figure 4: Generic Push Relabel

Bibliography

[1] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM (JACM)*, vol. 35, no. 4, pp. 921–940, 1988. 1

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009. 1