BLG 336E
Homework 3 Report
150130032 – Baran Kaya

Algorithm

I selected Karger Min-Cut algorithm for that project. This algorithm selects random edges and merges selected edges 2 nodes. While merging it handles with other edges of the selected nodes. At last there are only 2 nodes and a few edges between them. Final result of the algorithm is number of edges between these 2 nodes.

Karger Minimum Cut Algorithm

The algorithm works however; it randomly selects edges. Therefore, results can change with every run. With example.txt data in the project pdf results were 2 or 3. Success probability of the Karger algorithm is $\binom{n}{2}/(2^{n-1}-1)$ [n is number of edges].

Code - Main

Program first reads the file and saves every edges' shops in the pair vector. After that, it creates shop and road objects with pairs and stores them in ShopsVec and RoadsVec vectors. While adding new shops to the ShopsVec it checks if that shop is already created or not and if it is not in the ShopsVec then it adds new shop to the ShopsVec. Finally, main function calls the kargerMinCut function with ShopsVec and RoadsVec parameters and prints it result to the screen.

Code – Karger function

In kargerMinCut function, it copies the ShopsVec and RoadsVec because it will change them. After copying it selects the random edge (road) from RoadsVec and calls mergeNodes function with that index and copied ShopsVec, RoadsVec. It calls this function until there are only 2 nodes in the graph. Finally, it returns the number of edges between these 2 nodes.

Code – Merge nodes function

In mergeNodes function, it first finds randomly selected edge's 2 shops id (One of them is firstShop and the other one is secondShop, it will delete the firstShop). It deletes the selected edge (road) from RoadsVec. Then it searches these 2 shop ids in the ShopsVec and stores their indexes. After that, it starts adjusting ShopsVec. First, finds the shops that have a connection (road) to the firstShop and changes this shops road to firstShop to secondShop from its roadVec (firstShop will be deleted therefore it changes firstShop to secondShop -> Merging 2 nodes). Later it starts to adjusting the RoadsVec. For that purpose, it changes every firstShop in the RoadsVec to secondShop because firstShop will be deleted. Then it deletes the loop roads (edges) form the RoadsVec and finally it erases the firstShop from ShopsVec.

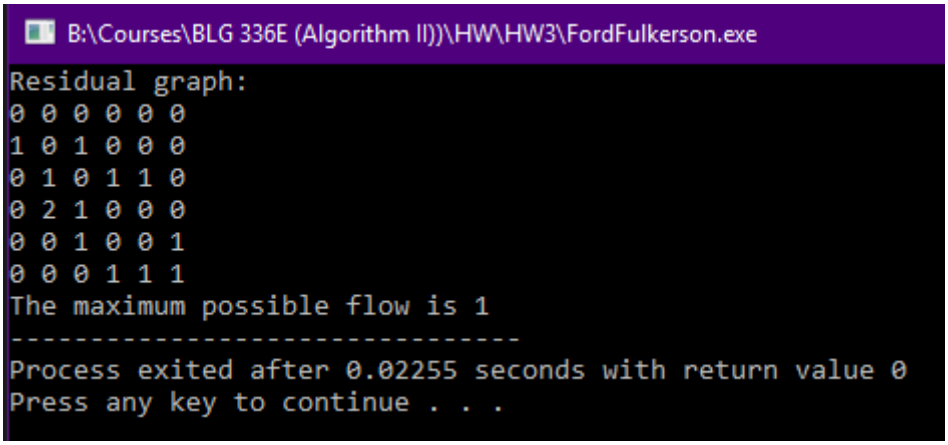
Different data structures

For better operations over object and classes Union-Find data structure can be used however; implementing it can be hard.

Other algorithms for min-cut problem

Although it is not the best algorithm for the minimum cut problem, it returns correct results sometimes (randomly). There are Karger&Stein and Parallelized Version algorithms for better results and calculation complexity but I could not find any understandable sources for these algorithms.

Ford Fulkerson Algorithm



The residual graph represents the max capacity that can flow over the edge of two nodes. Also they are used for calculating the max flow over those nodes/edges and finally it finds the total max flow over source to destination.

Compiling on SSH: g++ -std=c++11 150130032.cpp -o B --> It needs C++11.

150130032
Baran KAYA