CS4234: Optimisation Algorithms	Tutorial 5

Push-Relabel; More Max-Flow Application

V1.2: Steven Halim — *October 08, 2018*

Discussion Points

Q1: Please read `https://uva.onlinejudge.org/external/128/12873.pdf` and try to reduce this problem into a max flow problem, solve it using $O(n^2 \times m)$ Dinic's algorithm (assuming that you have such implementation ready), and analyze its time complexity.

Q2: Please perform a manual execution of a basic $O(n^2m)$ Push-Relabel algorithm works on the small flow network shown in Figure 1 (there is no VisuAlgo visualization on Push-Relabel algorithm yet but you can use other people's tool, like `http://www.adrian-haarbach.de/idp-graph-algorithms/implementation/maxflow-push-relabel/index_en.html`). For this question, you are allowed to perform the push or relabel actions *in any order* (but do not use the strategy mentioned in Q3 yet). Tutor may change the graph for the actual tutorial.

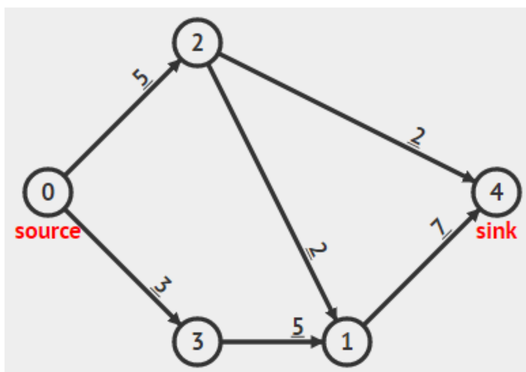

Figure 1: The initial flow graph

Q3: The 'slowest' part of a basic $O(n^2m)$ Push-Relabel algorithm is due to the $2n^2 + 4n^2m$ possible non-saturating push operations. We can make this bound tighter to $O(n^3)$ by doing this strategy: "If at each step, we choose the vertex with excess *at maximum height* (or in another word, we discharge *all* excess flow from that vertex first), then the number of non-saturating push operations throughout the algorithm is at most $4n^3$", thus giving rise to the tighter $O(n^3)$ Push-Relabel algorithm. In CLRS, this strategy is called the Relabel-To-Front version of Push-Relabel algorithm.

Now perform this strategy on the same Figure 1 and give a short sketch on why this is faster.

Past Paper Discussions

The tutor will then discuss some relevant questions from past papers.

Important

Please do not forget to hand in your 7 pages report for PS3 either before or after this T05.