Discussion Points

Q1: 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 Q2 yet). Tutor may change the graph for the actual tutorial.

Q2: 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.

Q3: Someone suggests that we can optimize the performance of Push-Relabel algorithm for Max-Flow problem by not processing vertices that still have excess (no more push or relabel operation on those vertices) when their heights are $\geq n$ if we only need the s-t Max-Flow value of the flow graph. Show that this idea is actually correct by explaining succinctly on what will happen to the excess flow in those non-processed vertices (i.e. vertices with heights $\geq n$) if we run Push-Relabel algorithm as per normal (i.e. until all vertices have no more excess flow)?

Q4: Back in Lecture 1, we have learned about the MIN-VERTEX-COVER (MVC) problem. In T03, we also have learned about the MAX-INDEPENDENT-SET (MIS) problem. We haven’t discuss the special cases of these two problems if they are asked on Bipartite Graph, so we will do it now.
Given a Bipartite Graph $G = (V_L, V_R), E$ of approximately 100 vertices and up to 2500 edges, show how to find the MVC and MIS on $G$ by reducing those problems into Bipartite Matching and further reduce them into Max Flow problems.

Follow up question: What if the MVC/MIS problems asked are the weighted variants?

Postscript

Next week in T08, we will talk about Graph Matching problems. We can solve (most) (Bipartite) Matching problems using any Max-Flow algorithm that you have learned so far or we can use a simpler $O$ algorithm that we will/(have) discuss(ed) in Lecture 08 (note that Lecture 08 comes before T08).