Girvan-Newman algorithm for community detection

M.T. Irfan

Main idea:

- Calculate the betweenness of each edge.
- Successively delete the edge(s) with the highest betweenness (and recalculate betweenness).

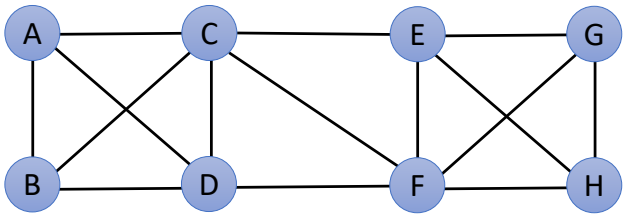
How to calculate the betweenness of an edge:

(Step 1) For each node A do the following:

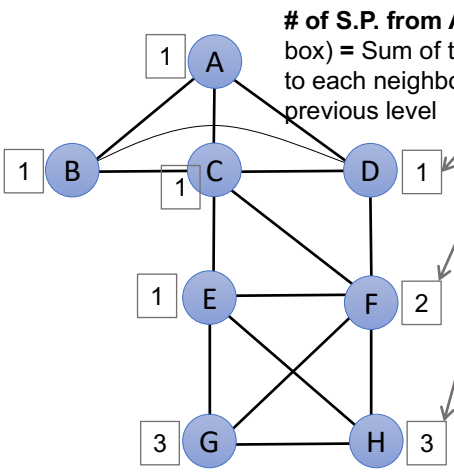
1. Do BFS starting with A.
2. Calculate the # of shortest paths (S.P.) from A to every other node.
   - # of S.P. from A to X = Sum of the # of S.P. from A to each neighbor of X in the previous level of BFS.
3. Calculate the quantity of water flow through each edge.
   - A sends 1 gal. to every other node X. Water from A to X gets evenly split across all S.P. from A to X.

(Step 2) Betweenness of an edge
= sum of all water flow through that edge (i.e., over all the BFS starting with each node).

Input network

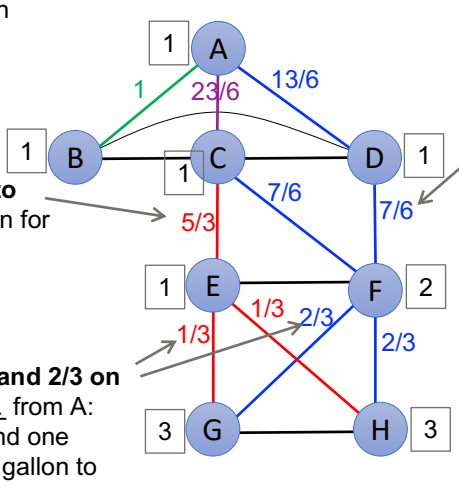


After BFS and #S.P. calculation, starting with A
(Note: showing all edges; it's not a BFS tree!)



# of S.P. from A to X (written in a box) **=** Sum of the # of S.P. from A to each neighbor of X in the previous level

Water flow calculation



Sanity check: Flow on AB + AC + AD = 1 + 23/6 + 13/6 = 7 gal. because there are 7 other nodes.

Why 5/3 gal. from C to E? E demands 1 gallon for itself plus 1/3 + 1/3 to send downstream. 1 + 1/3 + 1/3 = 5/3

Why 1/3 gal. on EG and 2/3 on FG? G has three S.P. from A: two S.P. through F and one through E. A sends 1 gallon to G. It's evenly split across the three S.P. to G.

Why 7/6 gal. from D to F? F demands 1 gallon for itself plus 2/3 + 2/3 to send downstream. This 1 + 2/3 + 2/3 = 7/3 gallons of water is split evenly across the two S.P. from A to F: through CF and DF.

Bottom-up calculation of water flow: Read this figure bottom up. Recall: A sends 1 gallon to every other node. Water flows only on S.P. to a node and splits evenly across all S.P. to it.

Final note: This is just **Step 1**. For each edge, we still need to sum up the water flow over each and every BFS starting with every node.