Weekplan: Graph Algorithms II

Philip Bille   Inge Li Gørtz   Eva Rotenberg

References and Reading

[1] Forster, Nanongkai, Yang, Saranurak, Yingchareonthawornchai: Computing and testing small connectivity in near-linear time. Sections 1 through 5.

Exercises

0 A polynomial is a polynomial. [w] Give a deterministic polynomial time algorithm for finding the minimum cut in a graph. Analyse its running time as a function of the number of vertices, n, and edges, m, in the graph.

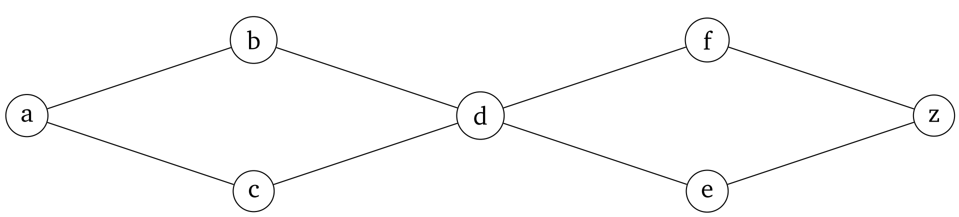
1 Heads or tails? Your friend Bianca Castafiore has tested all of her coins, and found one to be broken: "This coin always comes up tails when I toss it", Bianca says. She allows you to challenge this conclusion. Suppose the coin is unbiased and assume you may toss the coin i times. What is the probability that Bianca will gloat and say: "I told you, this was a broken coin!"?

2 Ford-Fulkerson-k In the video, it is said that using Ford-Fulkerson, we may either find a cut of size k separating u from v, or detect that no such cut exists, in time $O(m \cdot k)$. However, our textbook from previous algorithms courses say that Ford-Fulkerson runs in $O(m \cdot n)$ time on a graph with edge-capacities 1. Show how to modify Ford-Fulkerson to have the required behaviour.

3 Vertex-splitting

3.1 Consider the following graph. If all edges are bidirectional with capacity 1, what is the maximum flow from a to z?

3.2 Draw the resulting graph after the vertex splitting described in the paper/video. What is the maximum flow from a_{out} to z_{in}?



3.3 If G has m edges and n vertices, how many edges and vertices has the graph that results from splitting G?

4 Sampling edges In order to detect a balanced cut, the paper/video suggests that we sample pairs of edges, and hope that these belong to either side of the cut.

Professor Tournesol suggests an easier solution: In stead, we can sample pairs of vertices.

For $k = O(1)$, give an example of a graph with a k cut which is balanced in terms of volume, but not in terms of the number of vertices.

5 Reversing an edge In the unbalanced setting, assume we know there is a k-cut separating L from R where $v \in L$, and assume further that the volume of L is l.

In the paper/video, it is suggested that we in this case would run a long BFS or DFS search from v, choose a random edge, and reverse a path from that edge back to v.

Your friends Dupond and Dupont find that strategy to be utterly cumbersome, not to mention inefficient, and suggest the following replacement: Simply run a DFS from v for $l + 1$ steps, and reverse the path from the last edge back to the starting point.

Explain using an example why their strategy does not work.

6 A datafile You are given a data file containing a description of a graph: the i'th line contains the neighbours of vertex i, and the file name contains information on the number of vertices.

From its name, and the size of the file in kB, you can immediately see that there are 5000 vertices and between 20000 and 30000 edges.

Can you say anything about the vertex-connectivity of the graph?

7 Detecting a balanced cut Assume you are given a graph G and you have the further knowledge that all 42-cuts are *vertex-balanced* in the sense that either side of the cut contains at least 1% of the vertices.

Give a randomised algorithm that finds a 42-cut with 95% probability. What is its asymptotic running time?