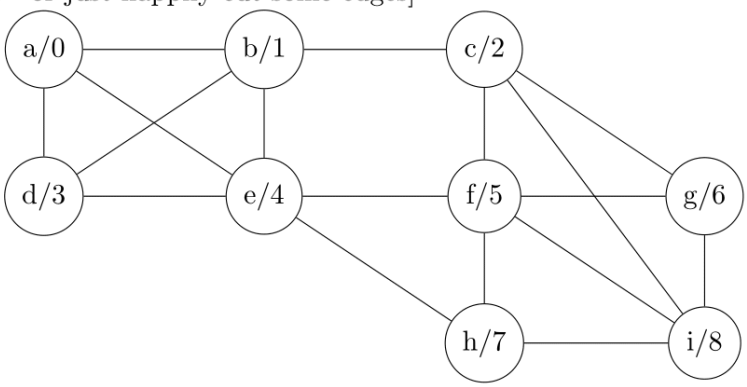
CSC 404 - Activity/Project 12 - Name: Chris Glanzer

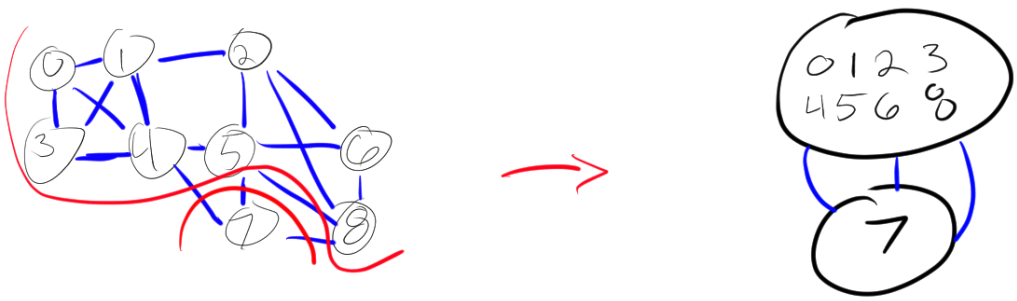
Problem 1. Consider the following graph/network – Give it a fun backstory :-). [Be the villain you were born to be - or just happily cut some edges]



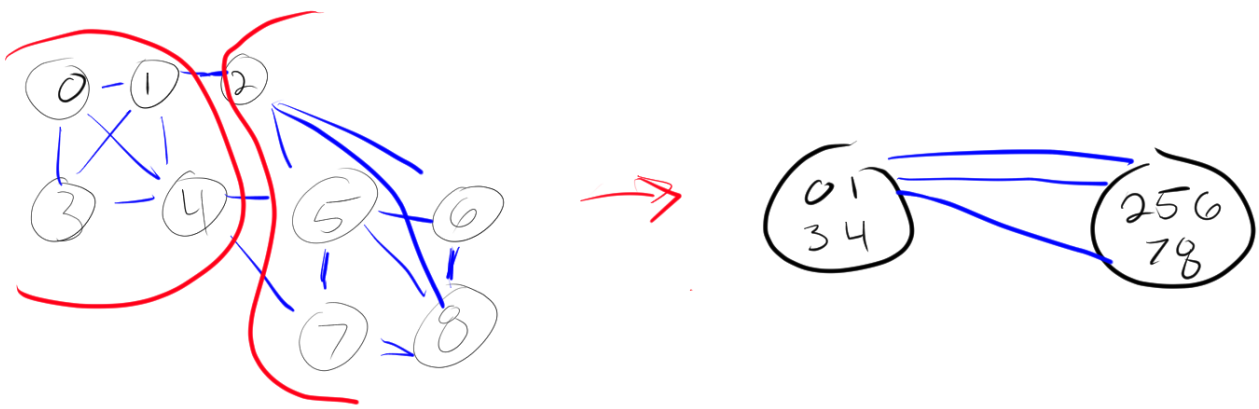
a. Run the Simple Algorithm (Karger's Algorithm) on this graph to determine an edge-cut for this graph.

by hand ok? → shortcut of defining the regions resulting from the cut

any?



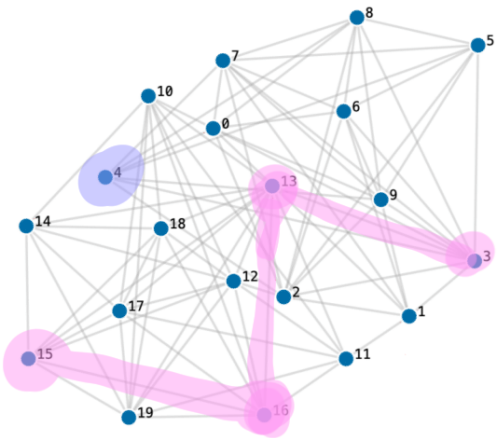
b. Run the Simple Algorithm (Karger's Algorithm) on this graph to determine another edge-cut for this graph.



c. What is the minimum (edge) cut size for this graph? (Did your run(s) come up with an edge cut with this size?)

3

Problem 2 (Bigger Networks!)**.** Consider the following graph/network ($n = 20$).



a. Run the Contraction/Simple Algorithm (you can use my Python implementation, but I encourage you to build your own – either improve my Python or build it in another lanuage) and record the number of cuts it returned to split the graph/network into two regions. What are the two regions and corresponding number of cuts?

R1: 3, 13, 16, 15

cuts: 30

R2: All the rest

b. Run the Contraction/Simple Algorithm $M = n^2 \ln(1000) = 20^2 \ln(1000) \approx 2763$ times and record the smallest cut observed (and two corresponding regions). This is (probably) the minimum cut – with probability of success of 99.999%.

R1: 4

cuts: 7

R2: All the rest

c. Run the Contraction/Simple Algorithm 10 times. How many of these runs resulting in the minimum cut? What about 100, 1000, 10000,... times?

Remark 1. The graph in Problem 2 has a special shape that I intentionally scrambled up (to make it look chaotic). If you look on D2L and view the AP12Graph2RealStory image you will very clearly see what is behind the scenes..

Problem 3. Currently we have been chatting about so-called 'edge'-cuts of a graph. That is, can we identified edges that once we remove them the graph is split into two (or more regions/components) – think destroying connections between nodes (if we are being evil).

What happens if we switch things up and chat about vertex-cuts? That is, can we identify vertices/notes that once we remove them (and the associated edged) the resulting graph is split into two (or more) components – think destroying nodes/cities (if we are being evil).

a. Jumping back to the graph/network on Page 1 - Identify a vertex-cut for this graph. What is the minimum vertex-cut size for this graph?

b. (Bonus) Ideas for algorithms/ideas for identifying vertex-cuts?