Last name _____

First name _____

LARSON—MATH 556–SAGE WORKSHEET 12

Network Flow

1. Log in to your Sage Cloud account.

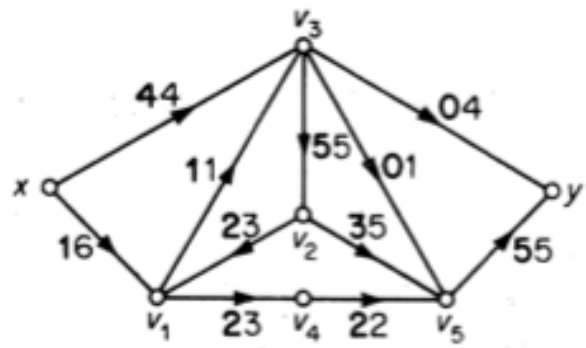
   (a) Start Chrome browser.

   (b) Go to `http://cocalc.com`

   (c) Click "Sign In".

   (d) Click project **Classroom Worksheets**.

   (e) Click "New", call it **s12**, then click "Sage Worksheet".



   x is the *source* and y is the *sink*. Each pair of numbers represents a *flow* f (darker) followed by a *capacity* c (lighter).

   Lets first code the underlying digraph d. There are 6 points. Let $x = v_0$ and $y = v_6$. Then we can represent the 7 points $v_0, v_1, \ldots, v_6$ with the list $[0, 1, 2, 3, 4, 5, 6]$.

2. Start by initializing a digraph with 7 points: `d=DiGraph(7)`.

3. Add the directed line from x to v_3 in the diagram (or from 0 to 3 in Sage's representation) with the command `d.add_edge(0,3)`. You can see how our digraph looks right now by evaluating `d.show()`

4. Add the rest of the directed edges to d. Use `d.show()` to check that your final digraph is correct.

5. Weights or capacities can be added to the lines as "edge labels". The capacity of the directed line from x to v_3 in the diagram (or from 0 to 3 in Sage's representation) is 4. We can add this capacity with the command `d.set_edge_label(0,3,4)`. To see how our network looks now use `d.show(edge_labels=True)`.

6. Now add the capacities to all the rest of the directed lines in the diagram. Use `d.show(edge_labels=True)` to check that your final network is correct.

Another way to input a network D is to use a *weighted adjacency matrix*. We will start with a 7×7 matrix A of zeros. Then for each directed line (i, j) we will represent the capacity of that line by setting $A[i, j] = c$. Note that A will not be symmetric. In general $A[i, j] \neq A[j, i]$.

7. Evaluate `A = zero_matrix(7,7)`. Then evaluate `A` to see what A looks like now.

8. Add the directed line from x to v_3 with capacity 4 in the diagram (or from 0 to 3 in Sage's representation) with the command `A[0,3] = 4`. Then evaluate `A` to see what A looks like now.

9. Now add the capacities to all the rest of the directed lines in the diagram. Check that your A matrix captures all the directions and capacities as the diagram.

10. Then evaluate `D=DiGraph(A,format='weighted_adjacency_matrix')` to get the network D itself.

11. To find the maximum flow between *any* pair of points i and j in a network D, evaluate `D.flow(i,j, value_only=True)`. To find the maximum flow between source x and sink y in our network D, evaluate `D.flow(0,6, value_only=True)`. What do you get?

    To find the minimum (weighted) cut between *any* pairs of points i and j evaluate `D.edge_cut(i,j,use_edge_labels=True, vertices=True)`. *use_edge_labels* tells Sage to use the capacities. *vertices = True* tells Sage to output the two sets S containing the source and T containing the sink that define the cut.

12. Evaluate `D.edge_cut(0,6,use_edge_labels=True, vertices=True)` to find the minimum cut between x and y in our network D. What do you get?

13. Prove that it *is* a min cut.