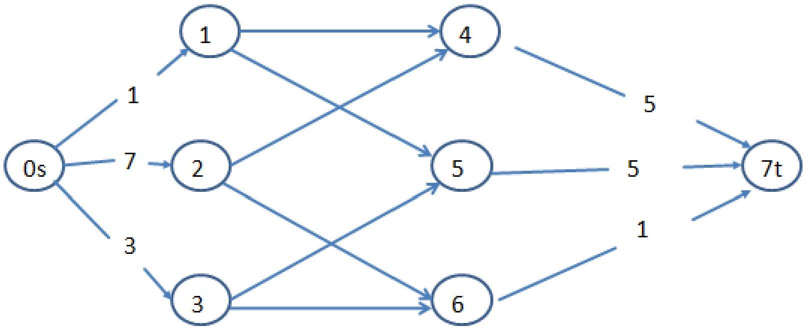
COS226 Maxflow Activity

1. Write the augmenting paths and determine the min-cut of the following FlowNetwork. Edges are labeled with their maximum capacity. Assume that edges that are not labeled have a maximum capacity of infinity.



2. Can the min-cut have vertices that are reached by backwards edges?

3. Use the table (from the assignment) below to determine if House Slytherin is already eliminated from the division. Draw the FlowNetwork and then calculate the augmenting paths of the Ford-Fulkerson algorithm.

```
                         w[i] l[i] r[i]        g[i][j]
i  team                  wins loss left   Gry Sly Rav Pac
-------------------------------------------------------
0  Gryffindor             83   71    8     -   1   6   1
1  Slytherin              80   79    3     1   -   0   2
2  Ravenclaw              78   78    6     6   0   -   0
3  Green Bay Packers      77   82    3     1   2   0   -
```

4. Then determine if the Green Bay Packers are eliminated? Is a graph needed? Why or why not?

5. How many game vertices are needed (at most) if there are N teams in your graph?

6. Show the contents of the array after one partitioning step in 3-way **string** quicksort of the following jelly bean flavors. Use the 3-way partitioning scheme (Algorithm 5.3 of Algorithms, 4th edition)

```
pear

lemon

wild grape

peach

plum

cherry

tangerine
```

7. What is MSD and LSD?

- We highly recommend that you use FlowNetwork.toString() to test and debug your program. The FlowNetwork.toString() for the Slytherin elimination problem is given below. Your numbering may vary, but the topology and weights should be the same.

  ```
  9 12
  0:  0->5 0.0/0.0
  1:
  2:  2->5 0.0/5.0
  3:  3->5 0.0/6.0
  4:  4->8 0.0/0.0  4->7 0.0/1.0  4->6 0.0/6.0
  5:
  6:  6->2 0.0/Infinity  6->0 0.0/Infinity
  7:  7->3 0.0/Infinity  7->0 0.0/Infinity
  8:  8->3 0.0/Infinity  8->2 0.0/Infinity
  ```

- After being passed to a FordFulkerson object, your FlowNetwork's flows will change to reflect the maxflow. The corresponding output of FlowNetwork.toString() is given below.

  ```
  9 12
  0:  0->5 0.0/0.0
  1:
  2:  2->5 5.0/5.0
  3:  3->5 1.0/6.0
  4:  4->8 0.0/0.0  4->7 1.0/1.0  4->6 5.0/6.0
  5:
  6:  6->2 5.0/Infinity  6->0 0.0/Infinity
  7:  7->3 1.0/Infinity  7->0 0.0/Infinity
  8:  8->3 0.0/Infinity  8->2 0.0/Infinity
  ```

- The core pieces of the API of our MaxFlow classes is given below, along with the internal representation of a FlowEdge.

  FlowNetwork:

  ```
  public FlowNetwork(int V) //empty graph with V vertices
  public void addEdge(FlowEdge e) //add edge e in both v's and w's adjacency lists
  public String toString() // string representation of Graph (excludes self loops)
  ```

  FordFulkerson:

  ```
  public FordFulkerson(FlowNetwork G, int s, int t) // max flow from s to t
  public double value() // return value of max flow
  public boolean inCut(int v) // is v in the s side of the min s-t cut?
  ```

  FlowEdge:

  ```
      private final int v;             // from
      private final int w;             // to
      private final double capacity;   // capacity
      private double flow;             // flow
  ```