Network Flow

Fundamentals of network flow

- Source -> Vertex **without incoming edges, generates flow**
- Target -> Vertex **without outgoing edges, collects flow**
- Edges -> The max amount of information that can be passed through(capacity)
- Flow -> How much information is currently passing through
- Flow Constraint Property -> Flow **CANNOT** be more than the capacity of an edge
- Flow Conservation Property -> **Incoming Flow == Outgoing Flow,** except for source and target

Maximum Flow Problem

1) Find total flow -> Identify total flow flowing out from source and flowing in to target(it will always be the same)

2) We want to find the maximum flow possible while adhering to the two properties stated previously

3) To tackle this, we run Ford-Fulkerson method

Ford-Fulkerson method

- Residual Network -> Graph that contains only residual edges and reversible edges
- Residual/Forward Edge -> Remaining capacity of a given edge
- Reversible/Backward Edge -> Flow that can be cancelled
- We find maximum flow using path augmentation
- Sum if multiple residual/reversible edge pointing to same direction, such sum equals edge capacity(this unit only works with simple graphs)

Complexity of Ford-Fulkerson

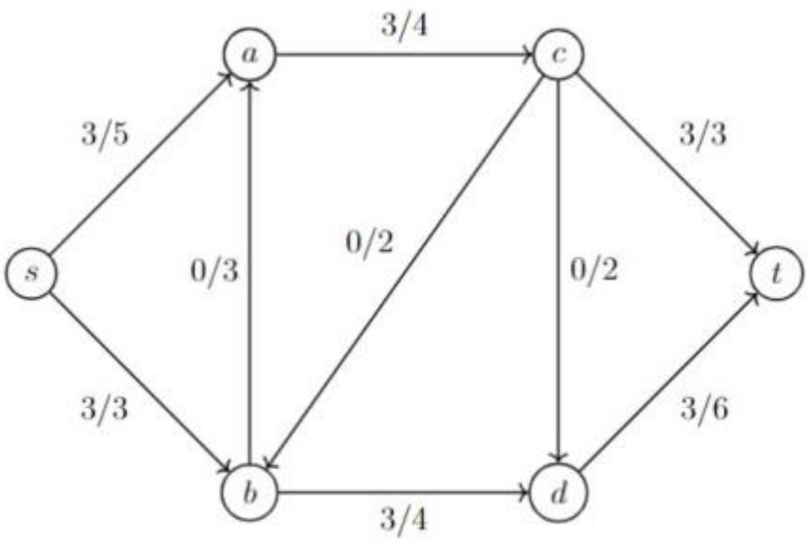
```python
def ford_fulkerson(my_graph):
    # initialzie flow
    flow = 0
    # initialize the residual network
    residual_network = ResidualNetwork(my_graph)
    # as long s there is a augmenting path
    while residual_network.has_AugmentingPath():
        # take the path
        path = residual_network.get_AugmentingPath()
        # augment the flow equal to the residual capacity
        flow += path.residual_capacity
        # updating the residual network
        residual_network.augmentFlow(path)
    return flow
```

- Initialize Residual Network -> O(E)
- Path Augmentation -> O(V+E) because we are using BFS/DFS to find path and augment
- Whole Process -> O(F) where F is the flow, as we keep iterating until we found the maximum flow possible
- Total -> **O(FE)**

<u>2022 Semester 1 Question 12</u>

Information

Consider the **flow network** below and answer the following questions.



Question 12

What is the **maximum possible flow** for the given flow network above? Just type the numerical answer.

The maximum flow is 7.

<u>Cut</u>

- Splitting / Cutting our network flow into two sections
- First section has the source vertex included, whereas the second section has the target vertex included
- Capacity of a cut -> Capacity of outgoing edges from the cut
- Flow of a cut -> Flow of outgoing edges – Flow of incoming edges

<u>Min-Cut Max-Flow theorem</u>

- Flow of cut <= Capacity of cut
- Flow of cut == Flow of network
- Capacity of min-cut = max-flow of a network
- Ford-Fulkerson terminates when there is a cut that meets the requirement of:
    1) Flow of each outgoing edge == Capacity of edge (no residual edges)
    2) Flow of each incoming edge to cut is 0 (no reverse edge)

Easiest way to find that cut is to find all reachable vertices from the source vertex in your completed residual network. All the reachable vertices from the source vertex are placed in the first section that has the source vertex included, the other vertices will be placed in the section that has the target vertex included.

<u>2022 Semester 1 Question 13</u>

Question 13

A cut partitions the vertices into two disjoint sets, *S* and *T*, where *S* contains all the vertices on the source side of the cut, and *T* contains all the vertices on the sink side of the cut.

Consider the **minimum cut** of the above flow network. Select the **vertices which are in** ***S*** from the list of vertices below.

The vertices are S and A.