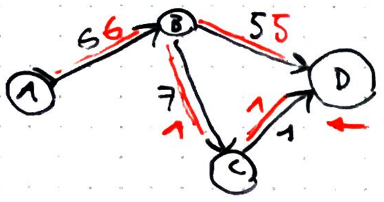
SW11

Maximum flow:

Directed Graph with positive weight function interpreted as capabilities.



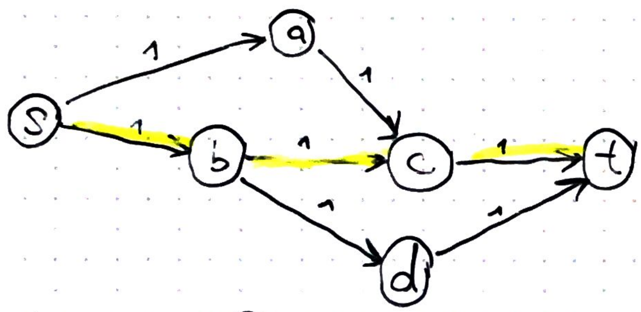
Goal is, to push as many flows as possible.

← limits the full connection to 1

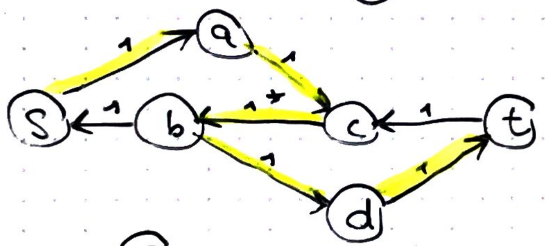
Total flow here would be 6. 5+1

Ford-Fulkerson algorithm:

Remember already used edges. Instead of removing them as possible edge, just draw it in reverse direction.

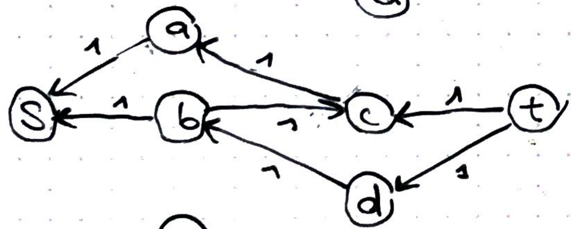


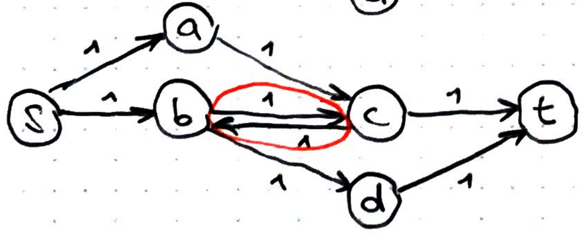
s → b → c → t capacity 1



S → a → c → b → d → t capacity 1

Don't think of going forward and back, just use this algorithm to find the total flow. In this case, it's get neutralized.



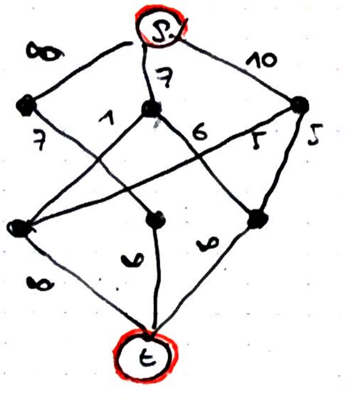


Result with combining all.

We don't use the red on at all.

Distribution Problem:

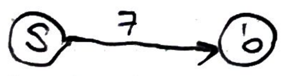
Just add s and t and weight them with ∞ or the total weight of outgoing edges.



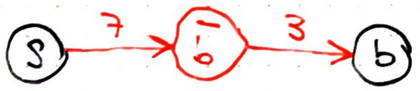
Use then the Ford-Fulkerson algorithm.

Modeling vertices with restrictions:

Add additional vertices before the restricted vertex and add new edge with the restriction weight.

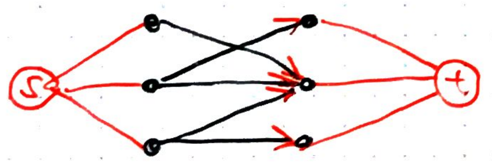


Now restrict 7 to 3



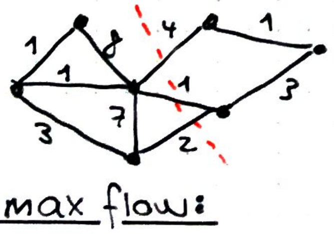
Biparite matching:

Add s and t and make the edges directed graphs.

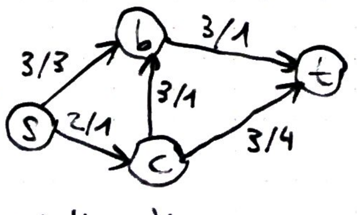


St - Cut:

Seperates the graph by a cut. It's value is the total of cutted edges.



value = 7



Min-cost-max flow:

With the Ford-Fulkerson, there can be multiple solution. You can now add costs to the edges and use this to find the best solution with Ford-Fulkerson respecting costs.