Network Flows

Syntax notes:

f(a,b): flow from a to b cap(a,b): capacity from a to b

Lemma: For any flow f and any cut (S,T), $size(f) \leq cap(S,T)$.

1. The flow across the cut $(S,T) \leq cap(S,T)$.
2. The flow across cut (S,T) = flow across cut (S-1, T+1).
3. The flow across cut({s}, V-{s}) = size(f)

Correctness of Ford-Fulkerson

Theorem: If residual network G_f has no augmenting path, then f is a max size flow.

Proof

Let S be the set of vertices reachable by directed path in G_f.

Let T=V S

All edges from S to T are saturated (and don't exist in the residual network). Can't reach from S to T via residual network.

This implies the flow across cut (S,T) = cap (S, T).

size(f) = f(S,T) = cap(S,T) because f(u,v) = cap(u,v) for all u in S, v in T

size of any flow $\leq$ cap(S,T).

If there is a cut where you can't get across via the residual network, that means it is the minimum capacity cut.

Max Flow - Min Cut Theorem

Size of the maximum flow = capacity of the minimum capacity cut.

Proof

Use proof of correctness of Ford-Fulkerson.

Integrality Theorem

If all capacities are integers then there exists a max flow such that every edge has integer flow.

Proof

Very simple proof by induction on number of augmentations of Ford-Fulkerson, since every iteration you add 1 unit of flow.

Maximum matching in a bipartite graph.

Matching

Matching in a graph is a set of edges in a graph such that no two edges in matching have a common endpoint.

Bipartite graph

Bipartite graph verticies can be partitioned into V_1 and V_2 so that for all edges, one endpoint is in V_1 and the other in V_2.

Can partition a square into a bipartite graph by coloring opposing corners.

Pentagons, not so much. Odd cycles are bad.

Maximum matching in a bipartite graph

How do you create a flow network through a bipartite graph such that it will give you the maximum matching?

You can take the bipartite graph and connect every node in each group to a vertex for that group. Set all edges to have capacity 1. Then calculate flow. Edges with flow > 0 are a maximum matching. Since every vertex has a maximum flow of 1 through it, there won't be any edges sharing a vertex.