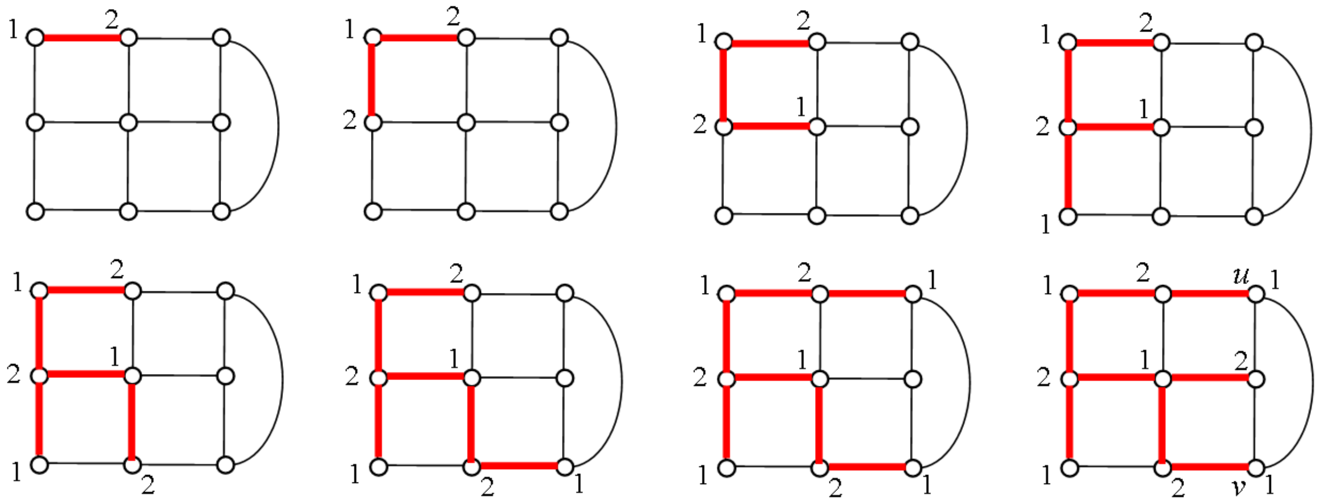
Week 8. Lecture 3

Alternating Paths: Recognising Bipartite Graphs

The problem of deciding if a graph is bipartite can be solved greedily. One way to see this is a nice adaptation of our old classic, the Maximal_Subtree algorithm. This is a slight digression (you will **not** be assessed on this adaptation) but it introduces a key idea in the theory of matchings.

Grow_Bipartition_Subtree(G)
 # Input: connected graph $G = (V(G), E(G))$
    Choose $v \in V(G)$; $T := (\{v\}, \emptyset)$ *# T is a tree with one vertex v and no edges*
    $X_1 := \{v\}$; $X_2 := \emptyset$ *# X_1 will be one partition for the bipartite graph, X_2 will be the other.*
    while there is an edge xy leaving T **do**
        $V(T) := V(T) \cup \{y\}$; $E(T) := E(T) \cup \{xy\}$ *# Growing the tree, as usual...*
        if $x \in X_1$ **then** $X_2 := X_2 \cup \{y\}$ **else** $X_1 := X_1 \cup \{y\}$ **fi**
    od;
    return(X_1, X_2)

Here is an example of how Grow_Bipartition_Subtree executes on a graph which is not bipartite:



Claim: If X_1, X_2 is the output of Grow_Bipartition_Subtree(G), then G is bipartite if and only if no edge of G has both endpoints in X_i, $i = 1, 2$.

We shall prove this using the following standard result (due to König, 1916; the 'if' part is not quite trivial):

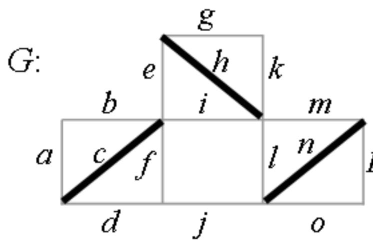
Lemma A graph G is bipartite if and only if every cycle has an even number of vertices.

Proof of Claim: For the 'if' part, suppose that no edge of G has both endpoints in X_i, $i = 1, 2$. Since G was assumed to be connected, Grow_Bipartition_Subtree will create a spanning tree. Then $X_1 \cup X_2 = V(G)$. Also $X_1 \cap X_2 = \emptyset$ since no vertex is assigned to an X_i more than once. So by definition G is bipartite.

For the 'only if' part, suppose that some edge joins two vertices x and y both belonging to X_1 or to X_2: let us assume it is X_1. Now there is a unique path P from x to y in the tree created by Grow_Bipartition_Subtree. By construction, P must alternate between X_1 and X_2. But P begins and ends in X_1, so it has an odd number of vertices. Then $P \cup \{xy\}$ is an odd cycle, so by the Lemma G cannot be bipartite. □

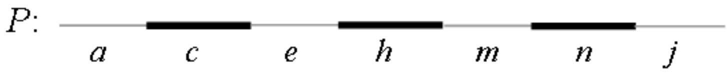
We can see this proof in action in the example above: vertices u and v both belong to X_1; and there is an alternating path on 7 vertices in the spanning tree which alternates between X_1 and X_2. Of course, this is not the shortest odd cycle but any odd cycle is enough to show the graph is not bipartite.

We shall use the idea of alternating paths to turn maximal matchings progressively into maximum matchings. Here is the maximal but non-maximum matching which we had earlier:



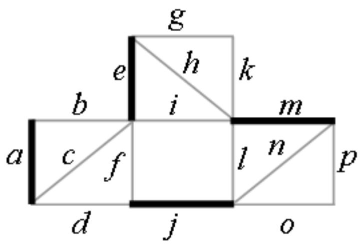
It is convenient to label the edges — the labels are not weights, they are just there for reference. Thus we can specify the matching as the subset of edges $M = \{c, h, n\}$.

We adapt the alternating path idea from **vertices** to **edges**: here is a path P which alternates between M-edges and non-M-edges:



Like the alternating path in our non-bipartite graph on the previous page, the path P has odd parity; this time it is the number of edges which is odd. And like the non-bipartite alternating path, P begins and ends with the same thing (a non-M-edge).

What if we swap M-edges and non-M-edges all along the path P: since P has odd length we will swap an even number of M-edges with a **larger** odd number of non-M-edges. *And our matching will get bigger:*



This extension of the matching works precisely because the end edges of P, that is a and j, both had an end-point not in M.

Definition: (1) A path in graph G is called ***M*-alternating** for a matching M if its edges alternate between M and $E(G)\backslash M$ (non-M);

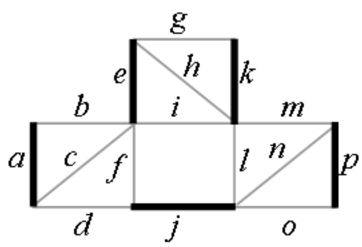
(2) A vertex v of G is called ***M*-unsaturated** for a matching M if it belongs to no edge of M;

(3) An M-alternating path in a graph G with matching M is called ***M*-augmenting** if its first and last vertices are M-unsaturated.

Theorem:[1] A matching M in a graph G is maximum if and only if no path in G is M-augmenting.

The 'only if' part of the proof of this theorem is easy: we have just seen that an M-augmenting path leads to a larger matching. The 'if' part is a bit more complicated and is left as an exercise.

To conclude our example, the 2nd version of our graph again has an M-augmenting path: edges p, m and k form such a path. Switching M and non-M edges on this path gives a perfect matching:



[1]Papadimitriou and Steiglitz attribute this theorem to C. Berge (1957) and R.Z. Norman and M.O. Rabin (1959). But Alexander Schrijver, who is very thorough about historical details, attributes it to Petersen in 1891, almost 60 years earlier!