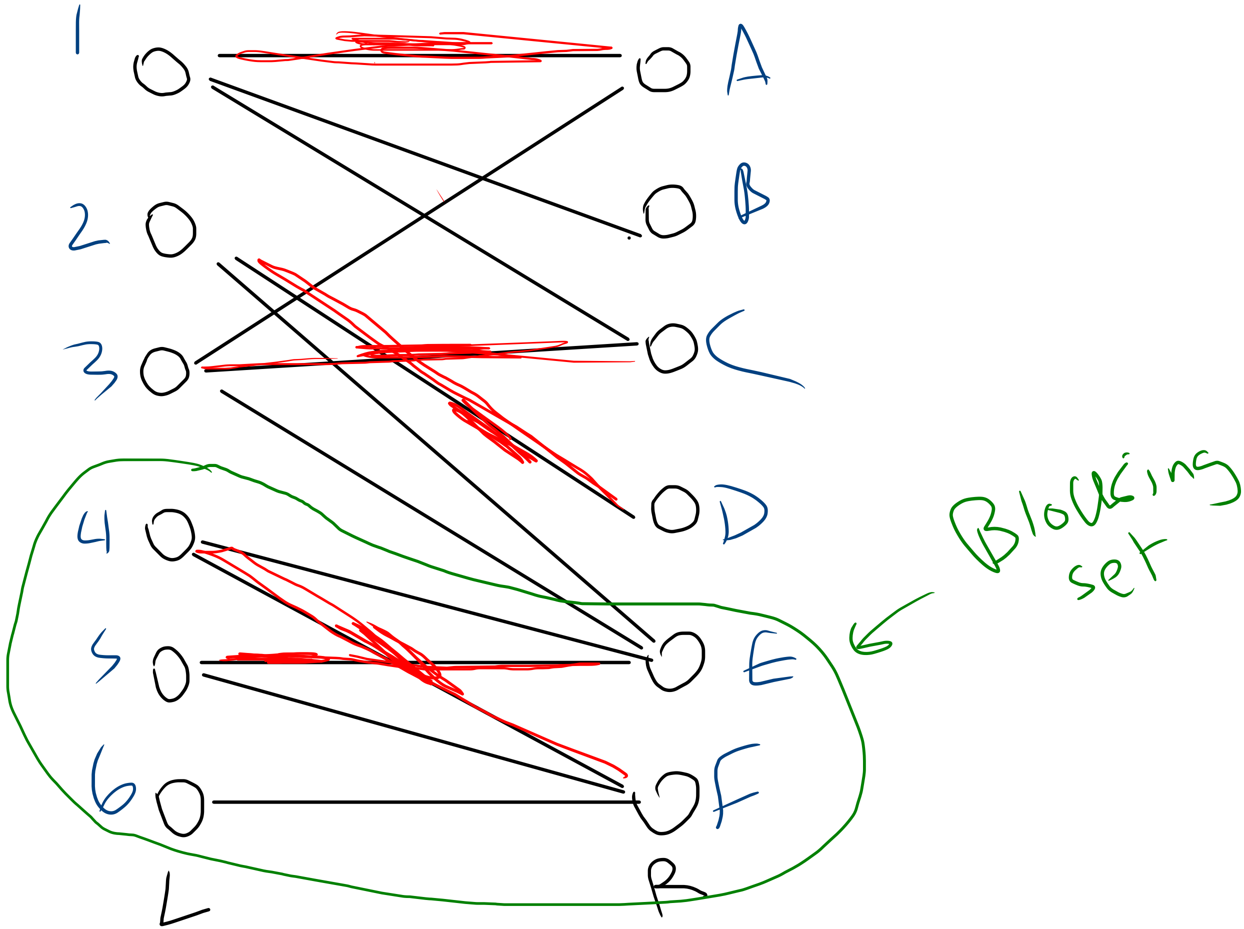
Bipartite Matching

Given bipartite graph, find a maximum-size <u>matching</u>.

<u>Matching</u>: A set M of edges which touches each node at most one.
(each node v is incident on at most one edge in M)


1
2
3
4
5
6
A
B
C
D
E
F
L
R
Blocking set

Applications : workers to jobs, tasks to machines, Goods to buyers, ...

Algorithms : - Using maximum flow techniques (in 270), can solve in $O(mn)$

- Improved $O(m\sqrt{n})$ (Hopcroft-Karp)

Perfect Matching: A matching which touches every vertex. In other words, each v is incident on exactly one edge of the matching.

Q: When does a perfect matching exist?

Hall's Marriage Theorem: A bipartite graph has a perfect matching if and only if every set of nodes S on one side has at least |S| neighbors.

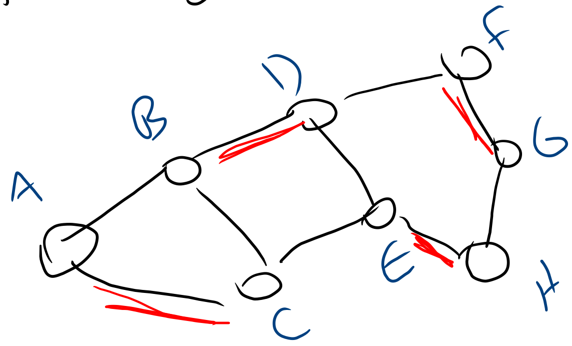
$$|N(S)| \geq |S| \quad \forall S \subseteq L$$

$$\forall S \subseteq R$$

neighbors

Non-bipartite Matching

The generalization to non-bipartite graphs.



Edmond's Blossom Algorithm: $O(mn^2)$

Improved algorithm by Micali & Vazirani: $O(m\sqrt{n})$

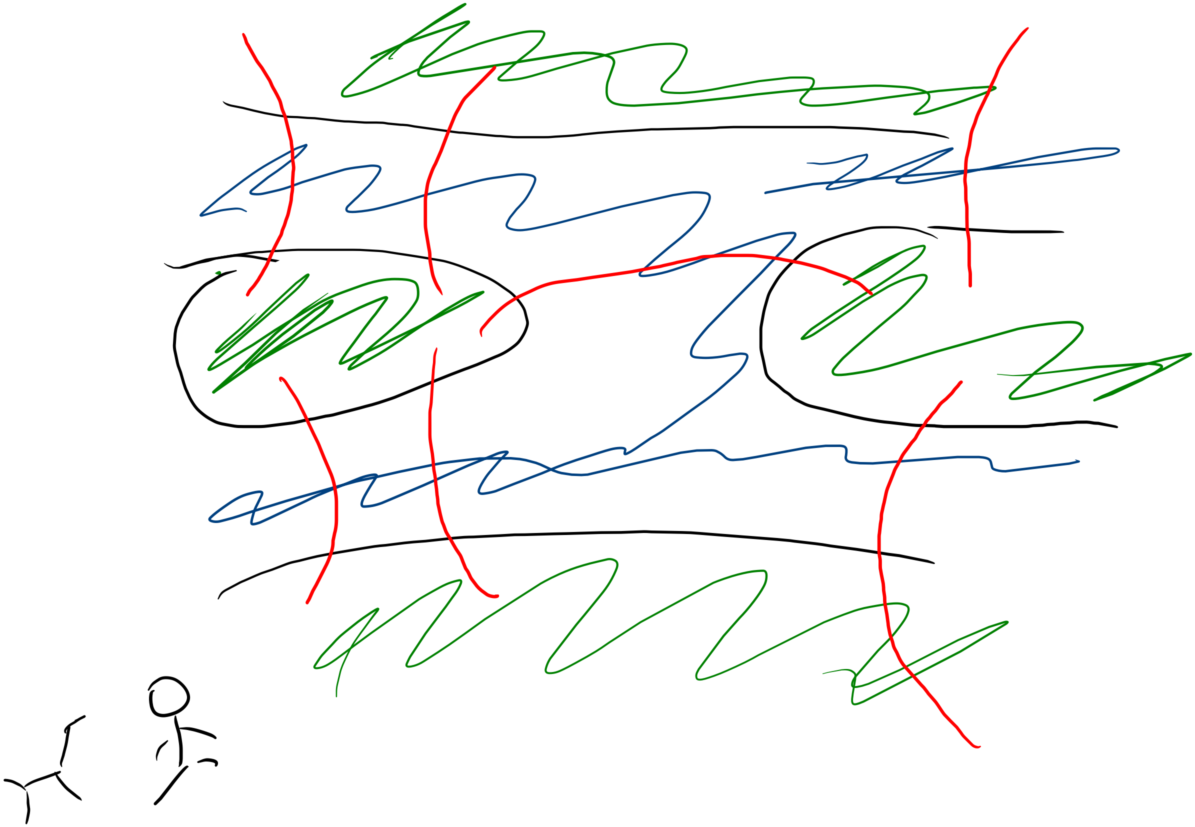
Hall's theorem no longer holds.
Complicated generalized condition does exist.

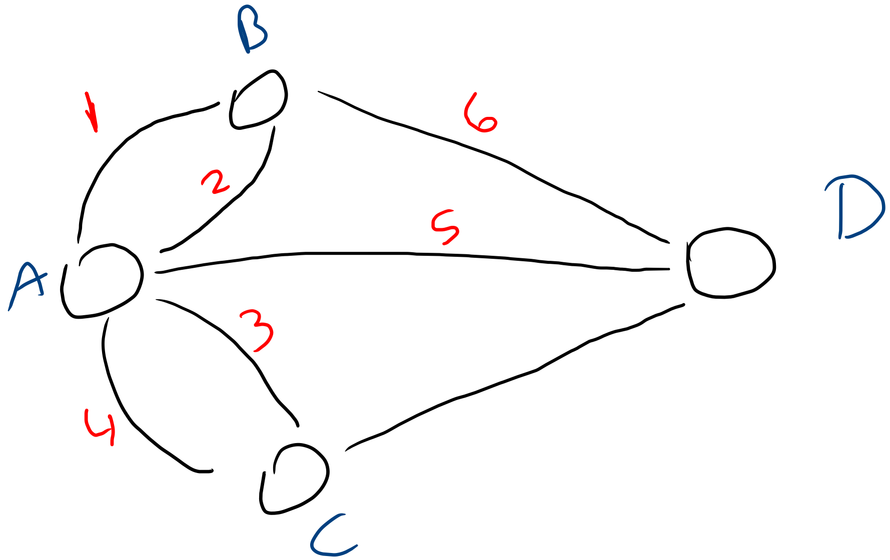
Eulerian Circuits

Given an undirected graph, find a walk that crosses every edge exactly once, and ends up where it started.

Eulerian Circuit

Historical Genesis: Bridges of Konigsburg

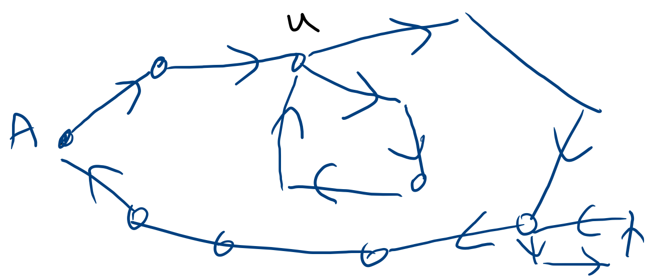



B
1
6
2
D
5
A
3
4
C

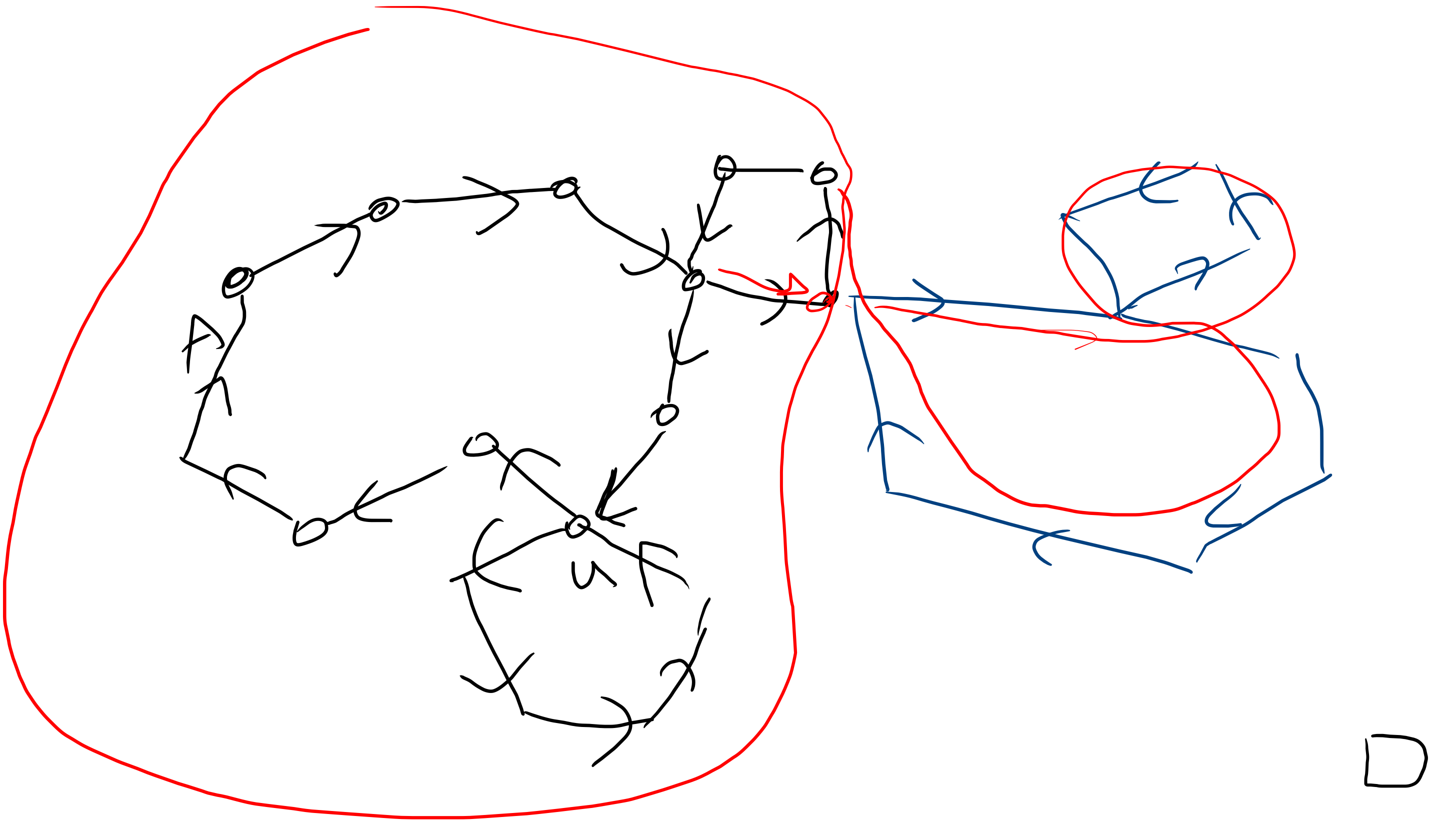
Theorem: A connected graph has an Eulerian circuit if and only if all degrees are even.

Proof Idea:

Eulerian circuit $\Rightarrow$ even degrees: Every time circuit enters it leaves on a different edge.



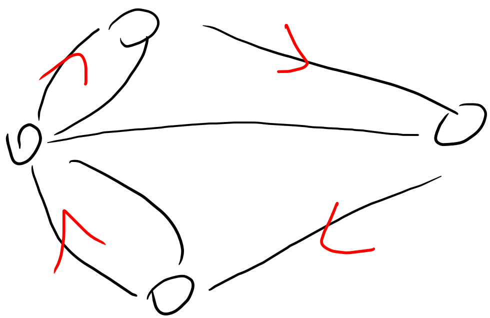
Even degrees → Eulerian Circuit :



The second part of the proof can be turned into an $O(m)$ time algorithm for finding Eulerian circuits whenever they exist.

Hamilitonian Cycles

Similar, but want to visit each node exactly once, Called a Hamiltonian Cycle.



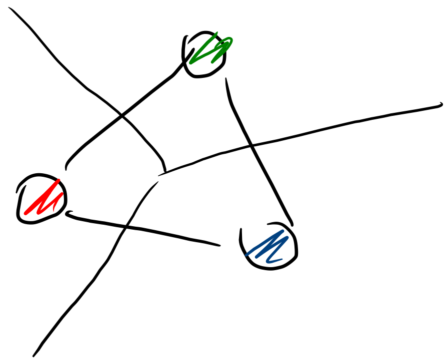
This problem is NP-Complete.
There is no nice characterization like the one for Eulerian Circuit.

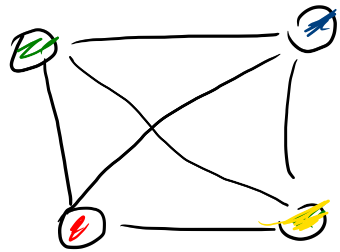
Graph Coloring

Given undirected graph G and k colors, assign colors to vertices so that all edges are bichromatic

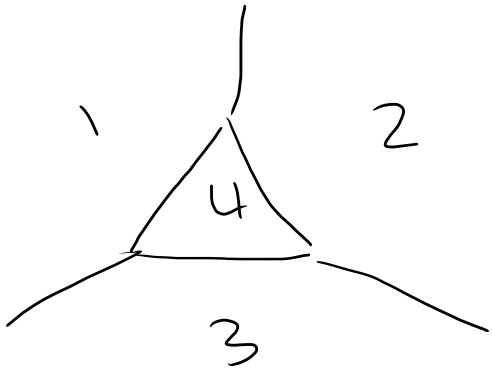
when k = 2 : Checking Bipartite .

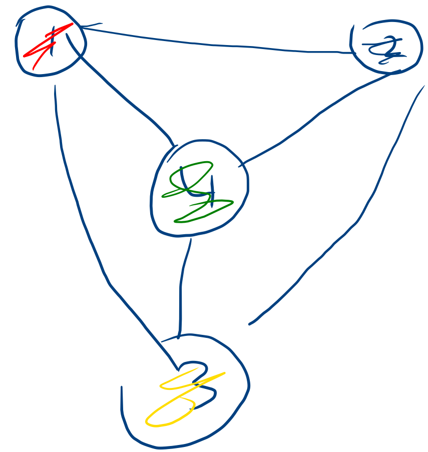
More generally: you want to color with as few colors as possible.





Application: Map Coloring





Map Coloring $\equiv$ Coloring planar graphs

<u>4-Color Theorem</u>: Every planar graph can be 4-Colored.

Soon followed by $O(n^2)$ Algorithm for Coloring planar graphs

For non-planar graphs, even 3-coloring is NP-Complete.

Chromatic # of a graph is minimum # of colors you need. NP-Complete to calculate for general graphs.