Lecture: Coloring

Shuai Li

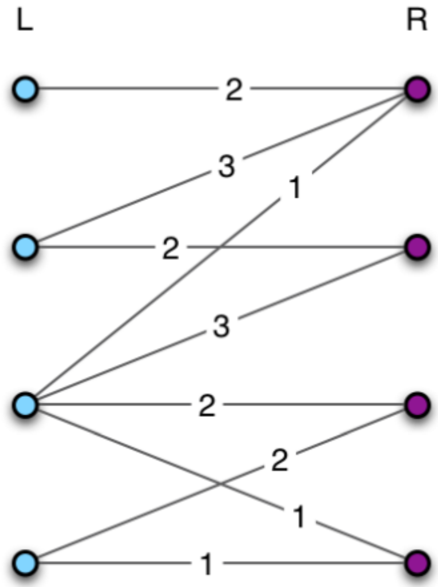
John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS445/index.html

Matchings

Hungarian example



- Theorem (3.2.11, W) The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover

Back to (unweighted) bipartite graph

- The weights are binary $0,1$
- Hungarian algorithm always maintain integer labels in the weighted cover, thus the solution will always be $0,1$
- The vertices receiving label 1 must cover the weight on the edges, thus cover all edges
- So the solution is a minimum vertex cover

Tutte's Theorem (TONCAS)

- Let $q(G)$ be the number of connected components with odd order
- Theorem (1.59, H; 2.2.1, D; 3.3.3, W)
  Let G be a graph of order $n \geq 2$. G has a perfect matching $\Leftrightarrow q(G - S) \leq |S|$ for all $S \subseteq V$

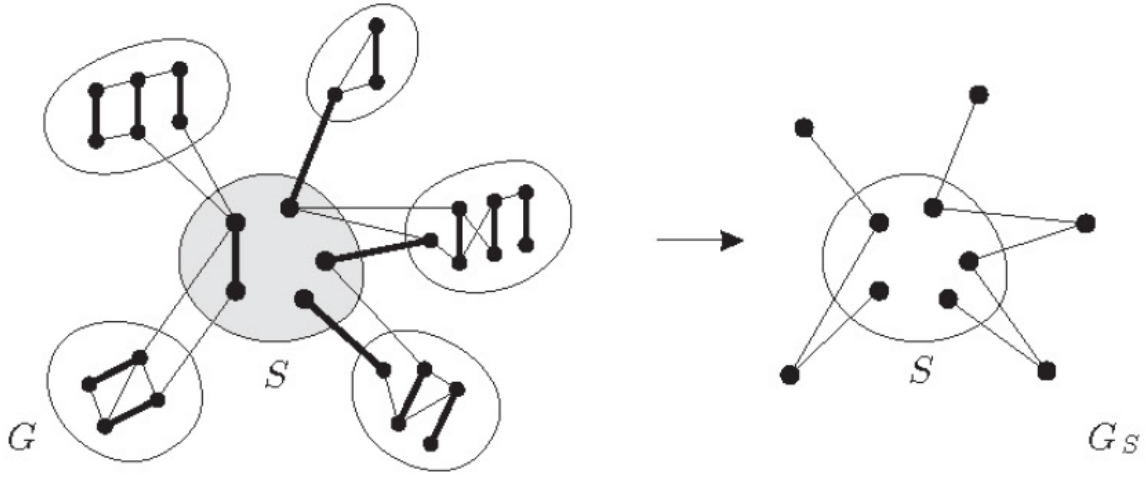
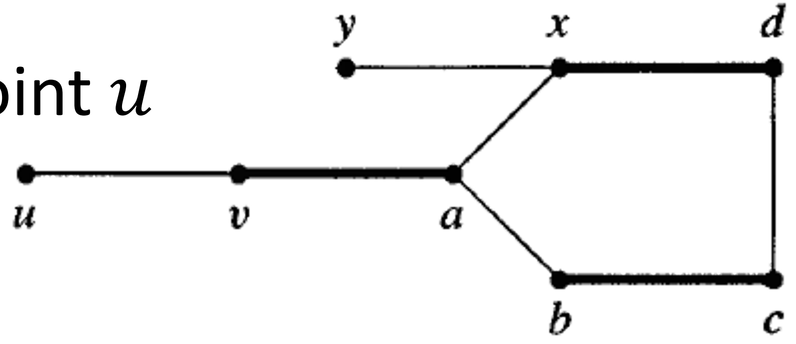


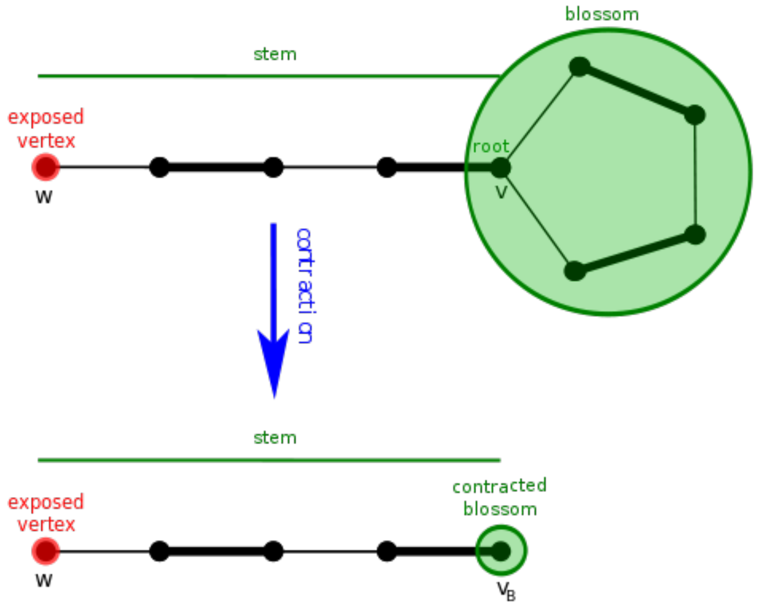
Fig. 2.2.1. Tutte's condition $q(G-S) \leqslant |S|$ for $q=3$, and the contracted graph G_S from Theorem 2.2.3.

Find augmenting paths in general graphs

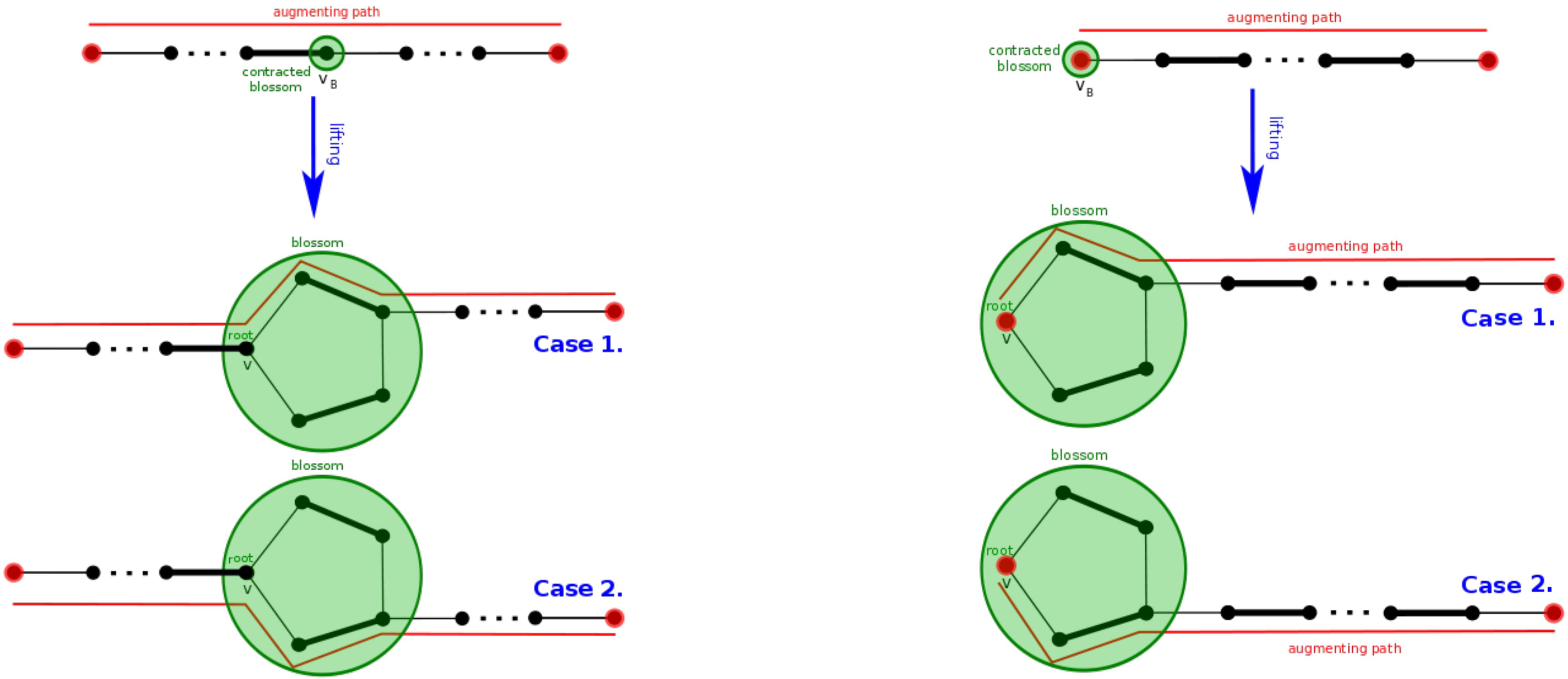
- Different from bipartite graphs
- Example: How to explore from M-unsaturated point u



- Flower/stem/blossom



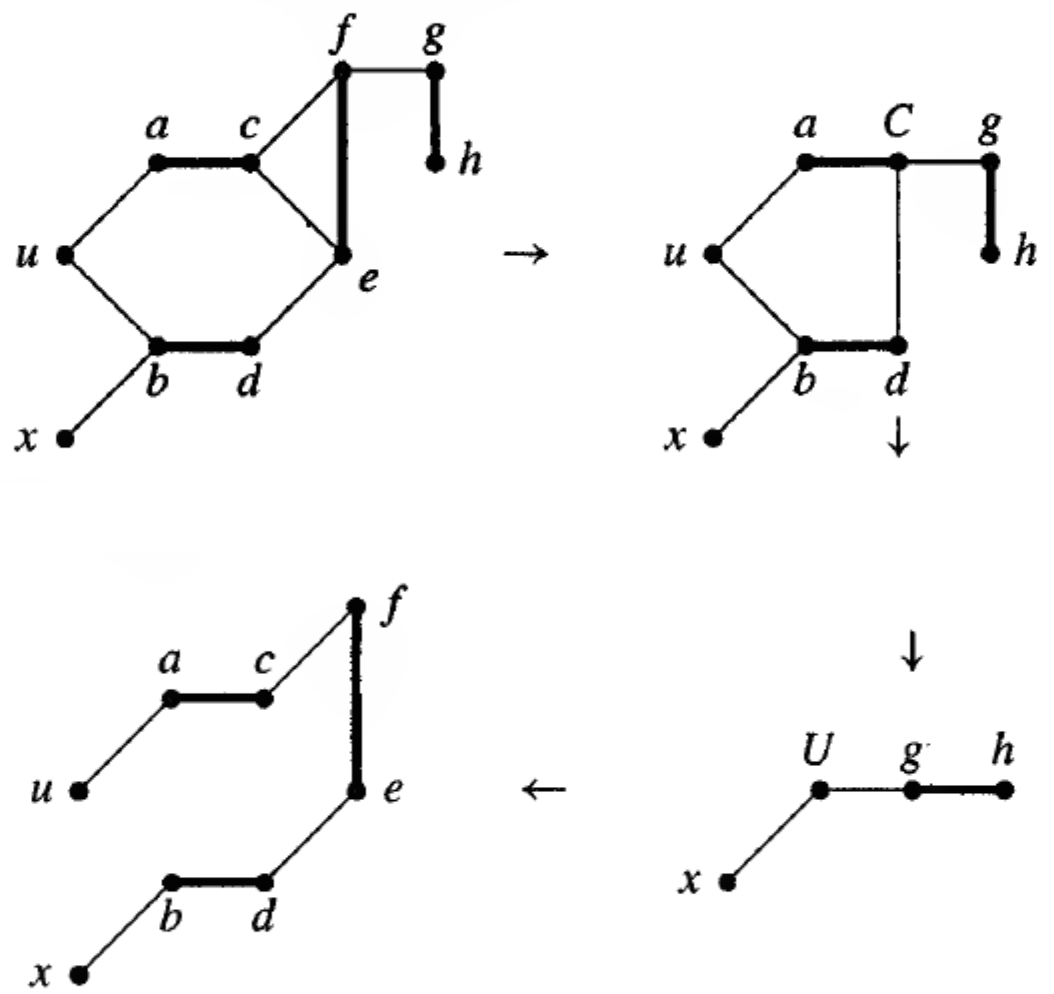
Lifting


augmenting path
contracted blossom
v_B
lifting
blossom
root
v
Case 1.
blossom
root
v
Case 2.
augmenting path
contracted blossom
v_B
lifting
blossom
augmenting path
root
v
Case 1.
blossom
root
v
Case 2.
augmenting path

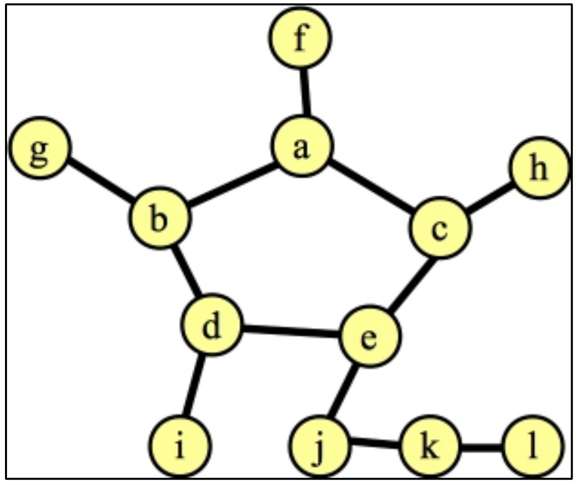
Edmonds' blossom algorithm (3.3.17, W)

- **Input**: A graph G, a matching M in G, an M-unsaturated vertex u
- **Idea**: Explore M-alternating paths from u, recording for each vertex the vertex from which it was reached, and contracting blossoms when found
  - Maintain sets S and T analogous to those in Augmenting Path Algorithm, with S consisting of u and the vertices reached along saturated edges
  - Reaching an unsaturated vertex yields an augmentation.
- **Initialization**: $S = \{u\}$ and $T = \emptyset$
- **Iteration**: If S has no unmarked vertex, stop; there is no M-augmenting path from u
  - Otherwise, select an unmarked $v \in S$. To explore from v, successively consider each $y \in N(v)$ s.t. $y \notin T$
    - If y is unsaturated by M, then trace back from y (expanding blossoms as needed) to report an M-augmenting u, y-path
    - If $y \in S$, then a blossom has been found. Suspend the exploration of v and contract the blossom, replacing its vertices in S and T by a single new vertex in S. Continue the search from this vertex in the smaller graph.
    - Otherwise, y is matched to some w by M. Include y in T (reached from v), and include w in S (reached from y)
  - After exploring all such neighbors of v, mark v and iterate

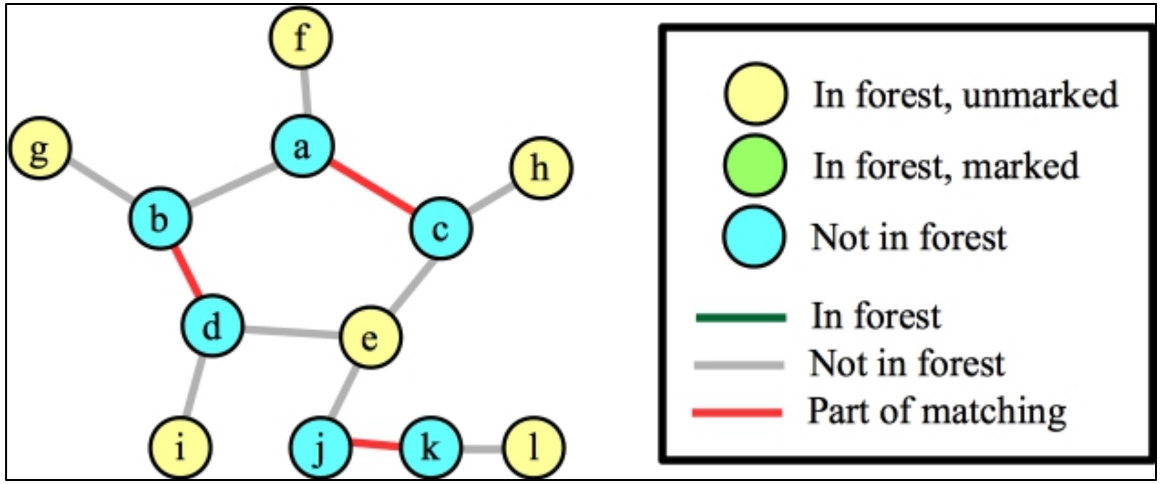
Example



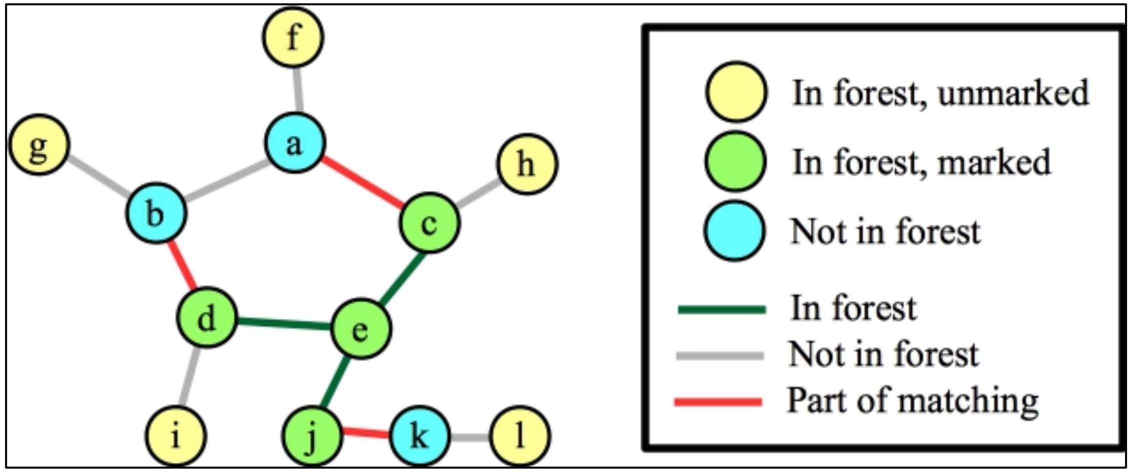
Example



f
g
a
h
b
c
d
e
i
j
k
l

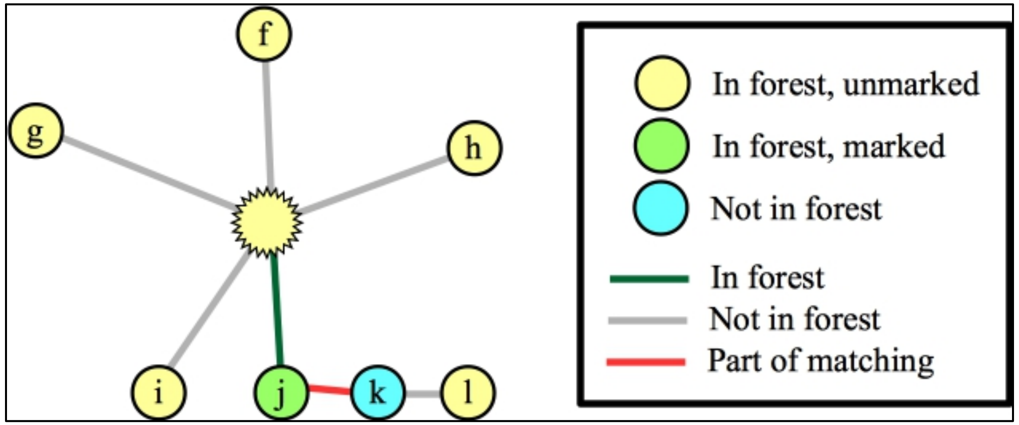


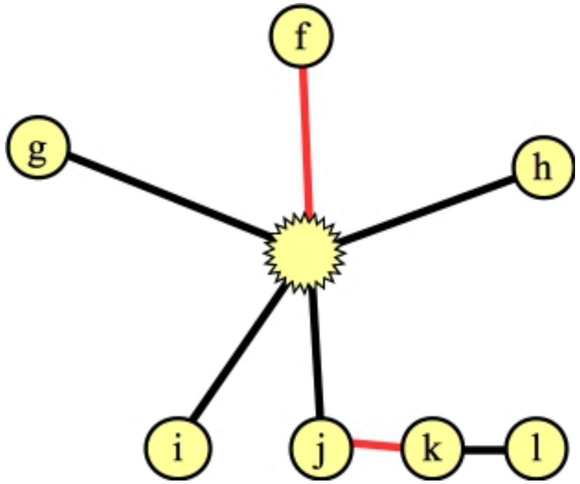
f
g
a
h
b
c
d
e
i
j
k
l
In forest, unmarked
In forest, marked
Not in forest
In forest
Not in forest
Part of matching

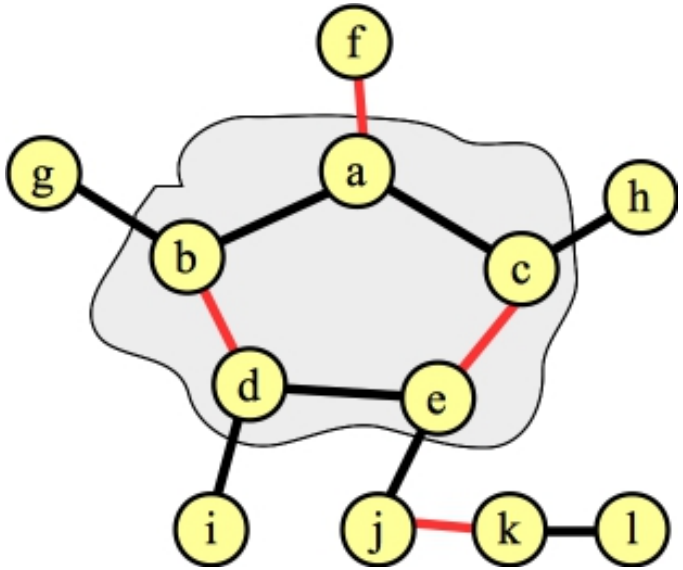


f
g
a
h
b
c
d
e
i
j
k
l
In forest, unmarked
In forest, marked
Not in forest
In forest
Not in forest
Part of matching

Example (cont.)



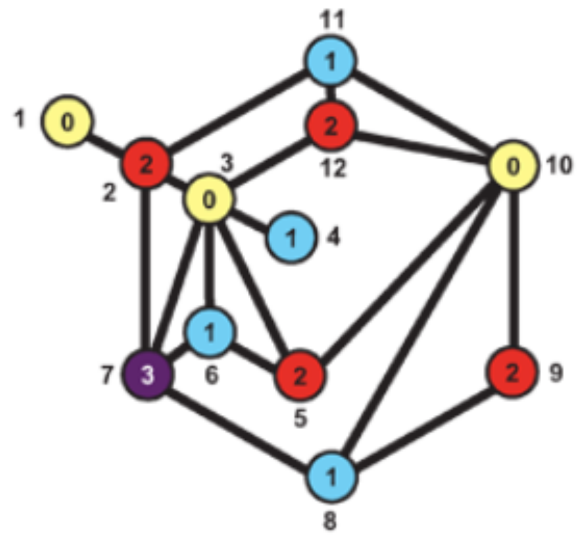




Coloring

Motivation: Scheduling and coloring

- University examination timetabling
  - Two courses linked by an edge if they have the same students
- Meeting scheduling
  - Two meetings are linked if they have same member



Definitions

- Given a graph G and a positive integer k, a k-coloring is a function $K\colon V(G) \longrightarrow \{1, \ldots, k\}$ from the vertex set into the set of positive integers less than or equal to k. If we think of the latter set as a set of k "colors," then K is an assignment of one color to each vertex.
- We say that K is a proper k-coloring of G if for every pair u, v of adjacent vertices, $K(u) \neq K(v)$ — that is, if adjacent vertices are colored differently. If such a coloring exists for a graph G, we say that G is k-colorable

Chromatic number

- Given a graph G, the chromatic number of G, denoted by $\chi(G)$, is the smallest integer k such that G is k-colorable
- Examples

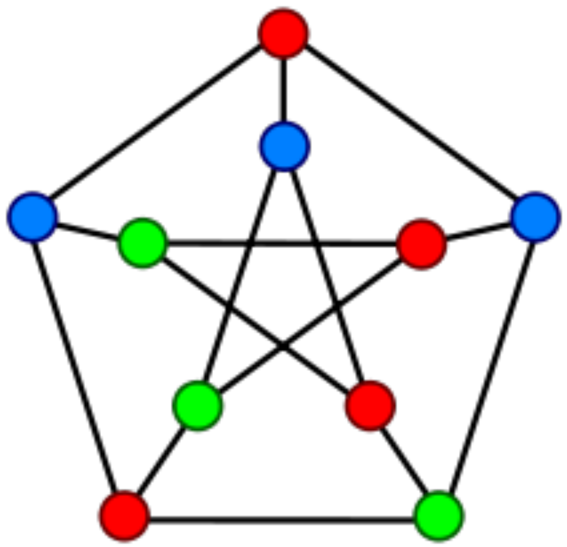
$$\chi(C_n) = \begin{cases} 2 & \text{if } n \text{ is even,} \\ 3 & \text{if } n \text{ is odd,} \end{cases}$$

$$\chi(P_n) = \begin{cases} 2 & \text{if } n \geq 2, \\ 1 & \text{if } n = 1, \end{cases}$$

$$\chi(K_n) = n,$$

$$\chi(E_n) = 1,$$

$$\chi(K_{m,n}) = 2.$$



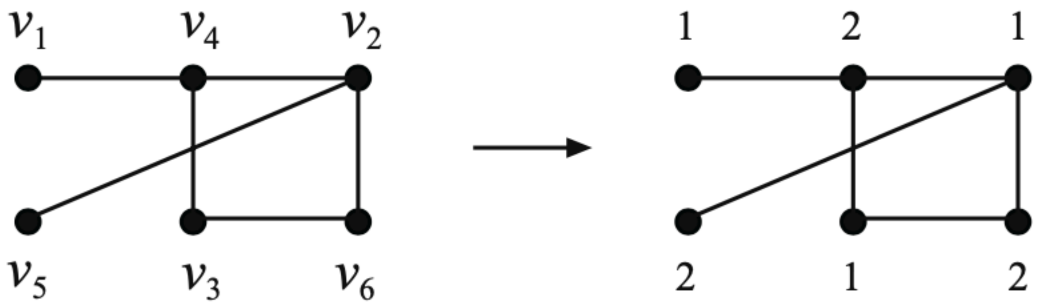
Bounds on Chromatic number

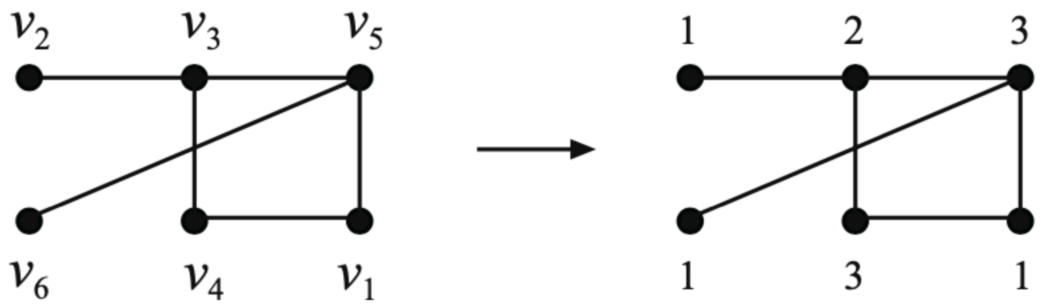
- Theorem (1.41, H) For any graph G of order n, $\chi(G) \leq n$

Greedy algorithm

- First label the vertices in some order—call them $v_1, v_2, \ldots, v_n$
- Next, order the available colors $(1,2, \ldots, n)$ in some way
  - Start coloring by assigning color 1 to vertex v_1
  - If v_1 and v_2 are adjacent, assign color 2 to vertex v_2; otherwise, use color 1
  - To color vertex v_i, use the first available color that has not been used for any of v_i's previously colored neighbors

Examples: Different orders result in different number of colors





Bound of the greedy algorithm

- Theorem (1.42, H) For any graph G, $\chi(G) \leq \Delta(G) + 1$
- The equality is obtained for complete graphs and cycles with an odd number of vertices

Brooks's theorem

- Theorem (1.43, H) If G is a connected graph that is neither an odd cycle or a complete graph, then $\chi(G) \leq \Delta(G)$

Chromatic number and clique number

- The clique number $\omega(G)$ of a graph is defined as the order of the largest complete graph that is a subgraph of G
- Example: $\omega(G_1) = 3, \omega(G_2) = 4$

- Theorem (1.44, H) For any graph G, $\chi(G) \geq \omega(G)$

Chromatic number and independence number

- Theorem (1.45, H; Ex6, S1.6.2, H) For any graph G of order n,

$$\frac{n}{\alpha(G)} \leq \chi(G) \leq n + 1 - \alpha(G)$$

The Four Color Problem

- Q: Is it true that the countries on any given map can be colored with four or fewer colors in such a way that adjacent countries are colored differently?
- Theorem (Four Color Theorem) Every planar graph is 4-colorable
- Theorem (Five Color Theorem) (1.47, H) Every planar graph is 5-colorable