Graph Coloring

CSE 373
Data Structures
Unit 16

Graph Coloring

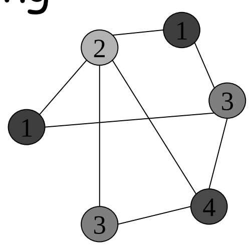
A problem that has lots of applications:

- Resource Allocation
- VLSI design
- Parallel computing

Definition: A coloring of a graph G(V,E) is a function c:Và N such that for any edge $(u,v) \in E$, $c(v) \neq c(u)$

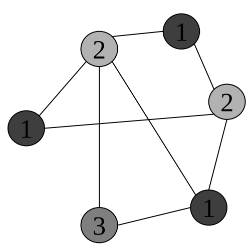
Graph Coloring

Example: coloring with 4 colors.



Problem: Given a graph G, color G using the minimal number of colors.

Example: same graph, 3 colors.



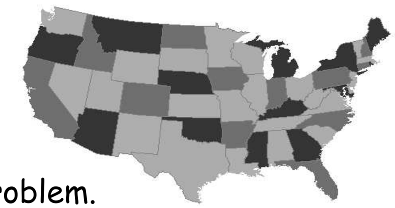
Definition: The chromatic number of a graph (denoted $\chi(G)$) is the minimal number of colors needed to color G.

The Map Coloring Problem

How many colors are needed in order to color a geographic map in such a way that neighboring countries get different colors?

The Map Coloring Problem

The map coloring problem can be reduced to finding the maximal chromatic number of a planar graph (a graph that can be drawn such that no two edges cross each other).



Vertices: the countries.

Edges: between neighboring countries.

For long, this was an open problem.

1852 – five colors are always enough, three is not enough (some maps require at least four colors).

1922 - four colors are enough for maps with at most 25 regions.

1976 - four colors are always enough.

Graph Coloring and Resource Allocation

Example: 9 groups of students are learning 5 courses in a quarter.

The course CSE501 is taken by groups 1,2,3

The course CSE502 is taken by groups 6,7

The course CSE503 is taken by groups 1,2,7,9

The course CSE504 is taken by groups 4,6,8

The course CSE505 is taken by groups 2,3,4,5

We want to schedule the exams such that no group will have more than one exam in one day, and the length of the exam period will be as short as possible.

Graph Coloring and Resource Allocation

Reduction to a coloring problem:

Vertices: courses.

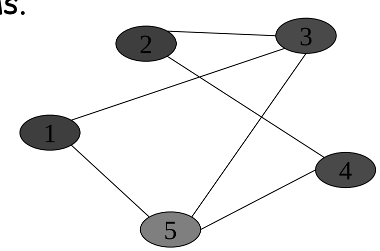
Edges: there is an edge between CSE_i and CSE_j if there is a group of students that needs to participate in both exams.

Possible solution:

Day 1: 501,502

Day 2: 505

Day 3: 503,504



Graph Coloring and Resource Allocation

Generally, given a resource allocation problem:

Vertices: processes that need resources.

Edges: between conflicting processes (that require a shared resource). This ensures that the two processes will not be scheduled simultaneously.

A coloring that uses k colors induces a partition of the processes into k phases.

Processes with the same color can be executed together – in the same phase.

2-colorable Graphs

Definition: A graph is k-colorable if it has a coloring with k colors.

Theorem: A graph is 2-colorable ⇔ it does not include a cycle of odd length.

Proof:

1. (⇒) Let G be colored with the colors 1,2. Assume that G includes a cycle of length 2j+1. W.l.o.g v_1 is colored with 1. It must be that for any even i v_i is colored 2, and for any odd i v_i is colored 1. Therefore, the two endpoints of (v_1-v_{2j+1}) are colored 1. A contradiction.
2. (⇐) Homework…

The Chromatic Number

Let $\Delta(G)$ be the maximal degree of a vertex in G.

Theorem: For any graph G, $\chi(G) \leq \Delta(G)+1$

Proof: Lets color the graph using at mast $\Delta(G)+1$ colors: Consider a list of the vertices in an arbitrary order. For each vertex in the list, determine c(v) to be the minimal integer which is not a color of any of the already-colored neighbors of v.

-This is a legal coloring: by definition, the color of v is different than the color of each of its neighbors.

- At most $\Delta(G)+1$ colors are used: when v is colored, at most $\Delta(G)$ neighbors of v are already colored.

The Chromatic Number

Remark: For some order of the vertices, this algorithm uses $\chi(G)$ colors.

Does it help us to find $\chi(G)$? practically, no!

We can check all the orderings, but this will take $O(n \cdot n!)$ which is a lot (next week we will define more formally why this is considered 'a lot').

Brook's Theorem: If G is not a complete graph nor a cycle of odd length, then $\chi(G) \leq \Delta(G)$.