RC7

Before start

hw5.5(ii)

Exercise 5.5 (4 pts) Define a relation $\preceq$ on $\mathbb{N}$ by

$$m \preceq n \Leftrightarrow (m = n) \vee (m \text{ even and } n \text{ odd}) \vee (m, n \text{ both even or both odd, and } m < n)$$

(i) (2 pts) Show that $(\mathbb{N}, \preceq)$ is a total order.

(ii) (2 pts) Sketch a Hasse diagram for $(\mathbb{N}, \preceq)$, and provide the explicit coordinates for all elements in $\mathbb{N}$.

Edge length of Hasse Diagram can vary.

To provide explicit coordinate, set the edge length as a converged series, like $\frac{1}{(2n+1)^2}$

Graph Homomorphism

Definition:

- simple graphs G and H
- a map from V(G) to V(H) which takes edges to edges

=> **nonedge can be mapped to anything**

=> There is an injective homomorphism from G to H (i.e., one that never maps distinct vertices to one vertex) if and only if G is a subgraph of H.

If a homomorphism $f : G \to H$ is a bijection whose inverse function is also a graph homomorphism, then f is a graph isomorphism

> *Definition in slides:*
>
> *Graphs G and H are isomorphic if there exists $f : G \to H$ and $g : H \to G$ such that $g \circ f = id_G$ and $f \circ g = id_H$,*

Note: Two graphs G and H are **homomorphically equivalent** if $G \to H$ and $H \to G$. The maps are not necessarily surjective nor injective.

Can you give an example that is homomorphically equivalent but not surjective or injective?

Graph Coloring

Theorem: A graph G is bipartite iff there exists a graph homomorphism $f : G \to K_2$

We can extend our theorem a little bit:

Theorem: A graph G is r-colorable ⇔ there exists a homomorphism from G to Kr

Corollary: A coloring of a graph G is precisely a homomorphism from G to some complete graph.

Firstly, let's defined r-colorable:

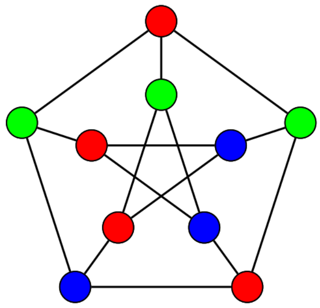
A proper coloring of a graph G is an assignment of colors to V (G) such that **no two adjacent vertices have the same color.**

A graph's **chromatic number** is the minimum number of colors needed to properly color the graph.

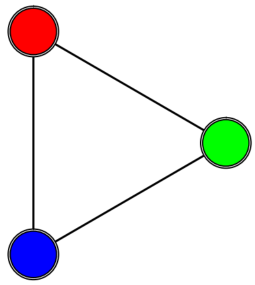
We say that a graph is n-colorable if it can be properly colored with n colors.

Now try to prove the theorem!

Example:

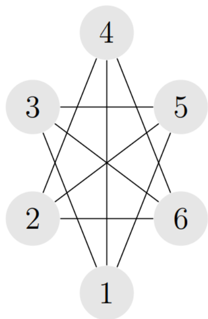


$\mapsto$



Example:

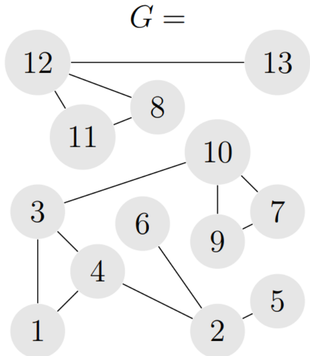
If we have 6 final periods with exams already scheduled such that no student must take consecutive exams, we can model this schedule as :



where each vertex is a final exam period and there exists an edge between vertices if a student can be scheduled to to take an exam in both time slots.(we want to ensure that no student is scheduled to take 2 exams in consecutive periods)

Consider the following exams, is it possible to assign a suitable schedule for them? (edge means there exists student taking both of them)

$G =$



Spanning Tree

T is a spanning tree of G

- subgraph
- tree = connected + without cycles
- V(T) =V(G) contain all vertices

Theorem: For connected graph with $|V(G)| \geq 2$, subgraph H is a spanning tree

$\Longleftrightarrow$ H is a minimal connected graph with V(T) =V(G)

$\Longleftrightarrow$ H is a maximal subgraph without cycles

Kruskal's Algorithm

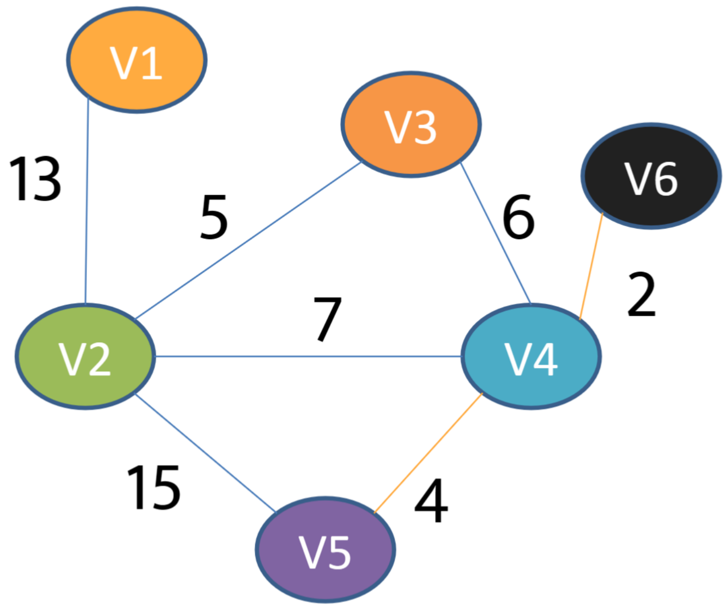
Find a minimum-cost tree

Greedy approach

- Maintain a "forest," or a group of trees /disjoint sets
- Iteratively select cheapest edge in graph
    - If adding the edge forms a cycle, don't add it
    - Otherwise, add it to the forest
- Continue until all vertices are part of the same set

Best for sparse graphs

Example:



Matroid

Spanning Trees v. Vector Space Bases

Finite dimensional vector spaces

A basis for a finite dimensional vector space is any of the following

- A minimal spanning/generating set.
- A maximal linearly independent set.
- Every element of the vector space is uniquely represented by as a linear combination of the basis vectors.

Finite connected graphs

A spanning tree is any of the following

- A minimal subgraph maintaining the same vertex set and connectedness.
- A maximal subgraph without cycles.
- For any two vertices, there is a unique path between them in the tree.

Actually they are the same structure: Matroid

There are many way to define the matroid, let's see the version of independent set:

A matroid, M, defined by independent sets comes with two pieces. We call the first piece, E, the ground set of M. E is a collection of elements.

The second piece is $\mathcal{I}$, the collection of independent sets of M. We write M = (E, $\mathcal{I}$). Elements of $\mathcal{I}$ are subsets of E satisfying some properties, and we call the elements of $\mathcal{I}$ independent.

1. $\emptyset \in \mathcal{I}$
2. If $A \in \mathcal{I}$ and $B \subset A$, then $B \in \mathcal{I}$
3. If $A \in \mathcal{I}$, $B \in \mathcal{I}$ and $|B| > |A|$, then there is $b \in B \backslash A$ with $A \cup \{b\} \in \mathcal{I}$

Example: Let E = {1, 2, 3, 4}. Let I = {∅, {1}, {2}, {3}, {4}, {12}, {13}, {14}, {23}, {24}, {34}}. In words, all the zero, one, and two element subsets of E are independent.

Example: Graphic Matroids (also known as cycle matroids of a graph). Let G = (V, E) be an undirected graph. Matroid M = (E, $\mathcal{I}$), where $\mathcal{I} = \{F \subseteq E : F \text{ is acyclic}\}$; i.e., forests in G.

click the links below if you want to learn more about Matroid

lec8.pdf (mit.edu)

Dijkstra’s Algorithm

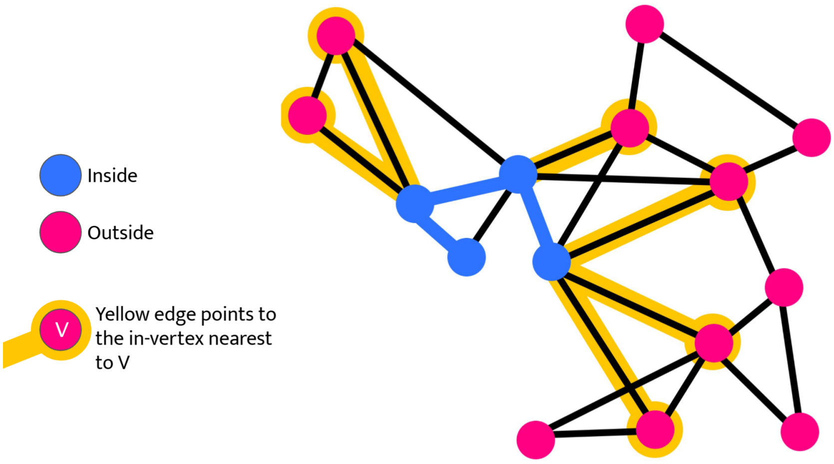
shortest path spanning tree for a certain vertex

Greedy Approach

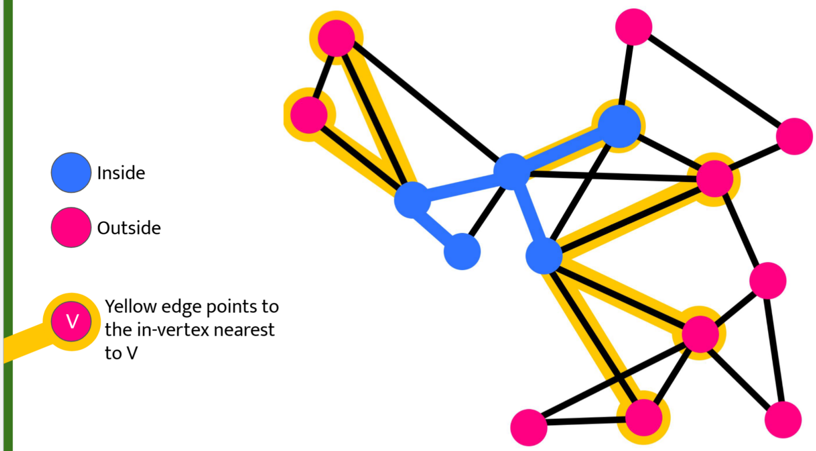
- Separate vertices into two groups:
  - “Innies”: vertices that are present in your partial MST at any point in time
  - “Outies” : the other vertices
- Iteratively add nearest outie, converting to an innie

Best for dense graphs

Example:

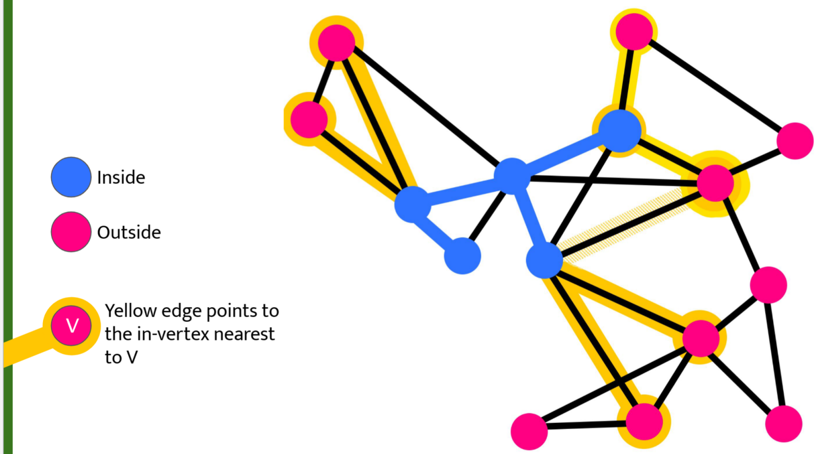


1 Find closest out-vertex **X**



1 Find closest out-vertex **X**

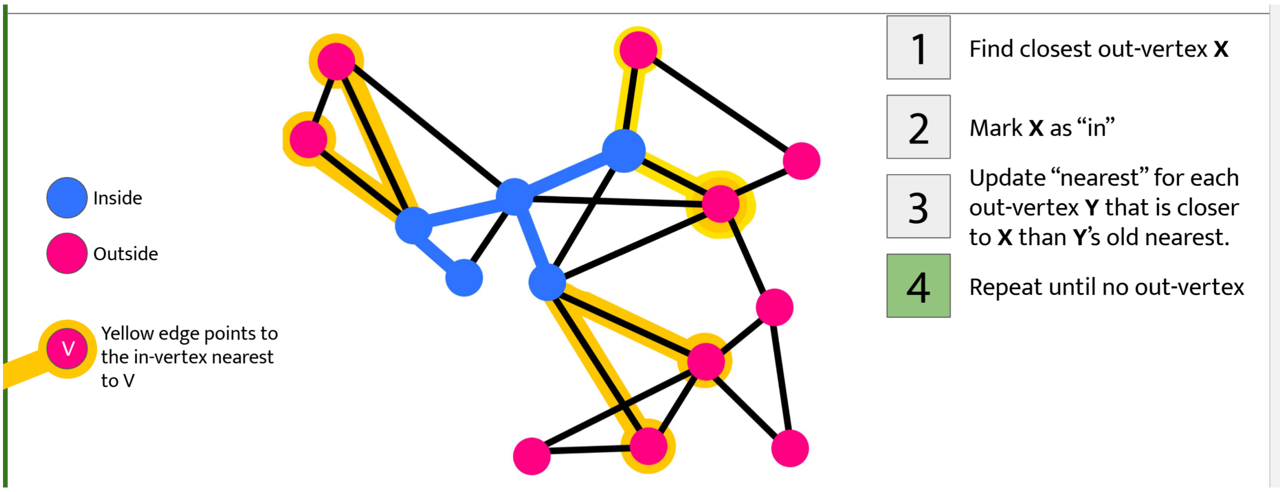
2 Mark **X** as “in”



1 Find closest out-vertex **X**

2 Mark **X** as “in”

3 Update “nearest” for each out-vertex **Y** that is closer to **X** than **Y**’s old nearest.



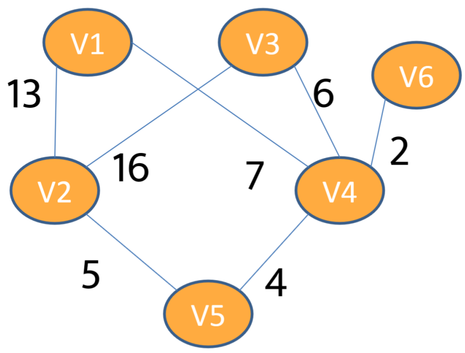
Detailed approach:

- Do the following until all vertices have been marked as "innies":
    - Choose the "outie" vertex X that is the smallest distance from any "innie"
    - Add X to the innie tree; keep track of its "parent" (closest innie), so you know what edge connects it to the tree
    - Update vertices connected to the new "innie" with new weights and a new parent index if this weight is smaller than their previous weight

inside MST?
how far from nearest innie?
which innie is closest?

v	k_v	d_v	p_v
V1	***F***	0	
V2	***F***	∞	
V3	***F***	∞	
V4	***F***	∞	
V5	***F***	∞	
V6	***F***	∞	

Example



Basic Number Theory

Division Algorithm Theorem: Let $n, d \in \mathbb{Z}$ with $d > 0$. There exists *unique* $q, r \in \mathbb{Z}$ such that

$$n = qd + r \qquad \text{and} \qquad 0 \leqslant r < d.$$

DEFINITION: Let $a, b \in \mathbb{Z}$. We say a *divides* b if there exists $q \in \mathbb{Z}$ such that $b = aq$. Equivalently, in this case, we say a is a *divisor* or *factor* of b, or that b *is divisible by* a, and write $a|b$.

Another way to prove:

Well-Ordering Principle: every non-empty subset of integers which is bounded below has a minimal element. Like most axioms, this is formalized "common sense."

How to use the Well-Ordering Principle to prove the Division Algorithm Theorem?

THEOREM 1.2: Let a and b be integers, and assume that a and b are not both zero. There exist $r, s \in \mathbb{Z}$ such that $ra + sb = (a, b)$.

The **Euclidean algorithm** is a method to find the GCD of two integers, as well as a specific pair of numbers r, s such that $ra + sb = (a, b)$. We will say that an expression of the form $ra + sb$ with $r, s \in \mathbb{Z}$ is a **linear combination** of a and b.[1]

Exercise: $524a + 148b = gcd(524, 148)$. Find a,b.

Reference

- Umich EECS 281 Lab9 slides
- Umich MATH 412 worksheet 1
- matroids
- davis-homomorphism-ups-434-2013
- lec8(mit.edu)