Math 180

Max Smiley

Winter 2022

1 Lecture 1

Well-Ordering Principle: *Any nonempty subset of the natural numbers* $\mathbb{N}$ *has a smallest element*

Def: a *set* is a collection of objects

- *sets are denoted with capital letters*
- *elements of sets are denoted with lower case letters*
- *if a set* X *contains an element* x*, then* $x \in X$

Def: an *unordered pair* containing x, y is the set $\{x, y\}$

- $\{x, y\} = \{y, x\}$

Def: an *ordered pair* containing x, y is the denoted as (x, y)

- $(x, y) \neq (y, x)$*, generally*
- *The pair* $(x_1, \ldots, x_n)$ *denotes the n-tuple of elements* $x_1, \ldots, x_n$

Def: the *empty set* $\emptyset$ is the set containing no elements

- *Note* $\{\emptyset\}$ *is a set with one element,* $\emptyset$

Def: the *cardinality* of a finite set X, denoted as $|X|$ is the number of elements it contains

2 Lecture 2

Def: a set X is a *subset* of another set Y if $x \in X \Rightarrow x \in Y$

- $\{3, 4, 5\} \subseteq \{3, 4, 5, 6\}$

Def: the *union* of two sets X and Y is defined as the set containing all elements of X and Y

- $X \cup Y := \{z : z \in X \text{ or } z \in Y\}$
- *For a collection of sets X_α indexed by some $\alpha \in I$, then*

$$\bigcup_{\alpha \in I} X_\alpha := \{\exists \alpha \in I : x \in X_\alpha\}$$

Def: the *intersection* of two sets X and Y is defined as the set containing elements only in both X and Y

- $X \cap Y := \{z : z \in X \text{ and } z \in Y\}$
- *For a collection of sets X_α indexed by some $\alpha \in I$, then*

$$\bigcap_{\alpha \in I} X_\alpha := \{x \in X_\alpha \ \forall \alpha \in I\}$$

Def: the *difference* of two sets X and Y is defined as the set containing all elements of X not in Y

- $X \setminus Y := \{z : z \in X \text{ and } z \notin Y\}$

Def: the *cartesian product* of two sets X and Y is the set of all ordered pairs (x, y) such that $x \in X$ and $y \in Y$

- $X \times Y := \{(x, y) : x \in X \text{ and } y \in Y\}$

Proposition 1.3.1: let X be a set containing 1, and the property $n \in X \Rightarrow n + 1 \in X$. Then $X = \mathbb{N}$

- *Prove via induction*

Induction (Theorem 1.3.2): Let $\{P(n)\}$ be a set of logical propositions. If $P(1)$ is true, and $P(n) \Rightarrow P(n + 1)$, then $P(n)$ is true $\forall n \in \mathbb{N}$

Def (1.4.1): a *function* from two sets A and B is a subset of $A \times B$ such that for any $a \in A$, there exists a unique $b \in B$ such that $(a, b) \in f$

- $f : A \to B \subseteq A \times B$ *s.t.* $\forall a \in A,\ \exists! b \in B$ *s.t.* $f(a) = b$

Def (1.4.2): let $f : A \to B$ and $g : B \to C$ be functions. The *composition of the functions* f *and* g is defined as a new function such that

- $(g \circ f)(a) : A \to C := g(f(a))\ \forall a \in A$

Def (1.4.3): the following define specific classes of functions for $f : A \to B$

- *injective:* $f(a_1) = f(a_2) \Rightarrow a_1 = a_2\ \forall a_1, a_2 \in A$
- *surjective:* $\forall b \in B, \exists a \in A$ *s.t.* $f(a) = b$
- *a function is bijective if it is both injective and surjective, it is also said to be invertible*

3 Lecture 3

Def: let $f : A \to B$ be a bijective function. We define the *inverse* of f as $f^{-1} : B \to A$ such that $f^{-1}(b) = a$, where a is the unique element such that $f(a) = b$

Def: let A and B be sets. A relation on A and B is a subset of $A \times B$

- *if R is a relation on $A \times B$, we say $x \in A$ is in relation to $y \in B$ if $(x, y) \in R$*
- *the notation $(x, y) \in R$ is equivalent to saying xRy*
- *all functions are relations, but not all relations are functions*

Def: let A, B, C be sets. Let $R \subseteq A \times B$ and $S \subseteq B \times C$ be relations. We define *the composition of relations of R and S as the relation $T = R \circ S \subseteq A \times C$ where $(a, c) \in T$ if $\exists b \in B$ s.t. $(a, b) \in R$ and $(b, c) \in S$*

Def (1.6.1): the following define specific classes of a relation R on a set X

- *reflexive: $xRx \ \forall x \in X$*
- *symmetric: $xRy \Rightarrow yRx \ \forall x, y \in X$*
- *antisymmetric: $(x, y) \in R \Rightarrow (y, x) \notin R \ \forall x \in X$ (for $x \neq y$)*
- *transitive: xRy and $yRz \Rightarrow xRz \ \forall x, y, z \in X$*

4 Lecture 4

Def: an *inverse relation* R^{-1} of a relation R, is the relation

$$R^{-1} := \{(b, a) : (a, b) \in R\}$$

Def: the diagonal relation on a set A, denoted Δ_A is the smallest reflexive relation on A

$$D := \{(a, a) : a \in A\}$$

Def: let R be a relation on a set A. We say R is an *equivalence relation* if it is *reflexive*, *symmetric*, and *transitive*

Def: let R be an equivalence relation on a set A. The *equivalence class* of an object $a \in A$, denoted as $[x]$ is the set of all $b \in A$ such that a and b form an equivalence relation

$$[a] := \{b \in A : (a, b) \in R\}$$

Def: an *equivalence* is an extension of the notion of equality to more general mathematical objects. Two objects are said to be equivalent if they belong to the same *equivalence class*

Def: a relation R on a set A is said to be an *ordering* of A if it is *reflexive*, *antisymmetric*, and *transitive*

- $\geq$ *is an ordering on* $\mathbb{N}$

Def: a relation R on a set A is said to be a *total ordering*, equivalently a *linear ordering*, if $\forall a, b \in R$, either aRb or bRa

- *Equivalently, if* R *is n ordering and* $R \cup R^{-1} = A \times B$

Proposition 1.6.3: *let* R *be an equivalence relation on a set* S*. Then*

- $\forall x \in X : [x] \neq \emptyset$
- *for two equivalence classes* $[a], [b] \in R$*, either* $[a] = [b]$ *or* $[a] \cap [b] = \emptyset$
- *if* R *and* S *are equivalences on* x*, then* $R[x] = S[x]\ \forall x \in X \Rightarrow S = R$

5 Lecture 5

Def: let $(X, \preceq)$ be an ordered set. We say x is an *immediate predecessor* of y if

- $x \prec y$
- *there exists no* $x' \in X$ *such that* $x \prec x' \prec y$
- *this relation is denoted as* Δ, *e.g.* $x \Delta y$ *iff* x *is an immediate predecessor of* y

Proposition 2.1.4: *let* $(X, \preceq)$ *be a finite ordered set, and* Δ *the immediate predecessor relation. Then* $x \prec y \Rightarrow \exists x_1, \ldots, x_n \in X$ *such that* $x_1 \Delta \ldots \Delta x_k$ *for* $k \geq 0$

- *in other words, we can take a finite number of "immediate predecessor steps" to go from* x *to* y *if* $x \prec y$

6 Lecture 6

Thm 2.2.1: *let $(X, \preceq)$ be a finite, partially ordered set. Then there exists a linear ordering on X, denoted $\leq$ such that $x \preceq y \Rightarrow x \leq y \; \forall x, y \in X$*

- *We can extend every partial ordering to a linear ordering*

Def: let $(X, \preceq)$ be an ordered set, and $a \in X$. We say a is a

- *minimal element of $(X, \preceq)$ if there is no $x \in X$ such that $x \prec a$*
- *maximal element of $(X, \preceq)$ if there is no $x \in X$ such that $a \prec x$*

Thm 2.2.3: *Every finite partially ordered set $(X, \preceq)$ has at least one minimal element.*

- *$\mathbb{Z}$, the set of all integers, for example, is an infinite set with no minimal element*

Def: let $(X, \preceq)$ be an ordered set, and $a \in X$. We say a is a

- *smallest element of $(X, \preceq)$ if $\forall x \in X : a \preceq x$*
- *largest element of $(X, \preceq)$ if $\forall x \in X : x \preceq a$*

Def: let $(X, \preceq)$ and $X', \preceq')$ be ordered sets. Let $f : X \to X'$ be a function. We say f is an *embedding* of $(X, \preceq)$ into $(X', \preceq')$ if

- *f is injective*
- $\forall x, y \in X : x \preceq y \Leftrightarrow f(x) \preceq f(y)$

Def: let $(X, \preceq)$ be an ordered set. Then there exists an embedding of $(X, \preceq)$ into the ordered set $(\mathcal{P}(X), \subseteq)$

Def: let $P = (X, \preceq)$ be an ordered set. Let A be a subset of X. We say A is independent in P if $x \preceq y$ for every distinct element $x, y \in A$

Def: let $P = (X, \preceq)$ be an ordered set. Let $x, y \in X$ such that $x \neq y$. We say x and y are incomparable if neither $x \preceq y$ or $y \preceq x$

- *equivalently, a set A is independent if all of its elements are incomparable*

Def: $\alpha(X)$ is the cardinality of the maximum independent subset of X

7 Lecture 7

Proposition 3.1.1: *let X be a set of n elements and Y be a set of m elements. Then the following hold*

- m^n *is the number of functions* $f : X \to Y$
- 2^n *is the size of the power set* $\mathcal{P}(X)$
- *If* $n \geq 1$, X *has exactly* 2^{n-1} *subsets of odd size, and* 2^{n-1} *subsets of even size*

Proposition 3.1.2: *let X be a set of n elements and Y be a set of m elements. Then the number of injective functions is*

$$\{f : X \to Y\} = \prod_{i=0}^{n-1}(m - i)$$

- *For example, we can treat the set of four letter words with distinct elements as an injective function from "four slots" to the 26 letters*
- *Then, the number of four letter words with distinct letters are* $26 \times 25 \times 24 \times 23 = 358800$

Def (3.1.3): let X be a finite set and $f : X \to X$ a function. We say f is a permutation of X if it is a bijection

- *for a set* $\{1, 2, 3\}$, *we can say* $\{2, 3, 1\}$ *is a permutation*
- *if* $|X| = n$, *the number of permutations (or bijective functions from X to X) is* $n!$
- $0! = 1$ *and* $n! = n(n-1)!$

Def (3.3.1): let $k, n \in \mathbb{N}$ such that $0 \leq k \leq n$. The *binomial coefficient* $\binom{n}{k}$ is the number of k element subsets of n

$$\binom{n}{k} := \frac{n!}{k!(n-k)!}$$

Proposition 3.3.2: *The following are properties of binomial coefficients*

- $\binom{n}{n-k} = \binom{n}{k}$

- $\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$
- $\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}$

Binomial Theorem (Theorem 3.3.3): For any nonnegative integer n, we have

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

- *We are essentially saying at each step, "how many different arrangements of k x's and $n-k$ y's can we procure?"*

Multinomial Theorem (Thm 3.3.4): For any nonnegative integer n, and real numbers $x_1, \dots x_m$, we have

$$(x_1 + \cdots + x_m)^n = \sum_{k_1 + \dots k_m = n} \binom{n}{k_1, \dots, k_m} x_1^{k_1} \cdot \cdots \cdot x_m^{k_m}$$

Note that

$$\binom{n}{k_1, \dots, k_m} := \frac{n!}{k_1! \cdot \cdots \cdot k_m!}$$

8 Lecture 8

Def: let $P = (X, \preceq)$ be an ordered set, and $A \subseteq X$. We say A is a *chain* if each two elements of A are comparable, or equivalently, A is a linearly ordered subset in P.

Def: $\omega(P) :=$ maximum number of elements of a chain in P

Thm 2.4.5: *let $P = (X, \preceq)$ be a finite ordered set. Then $\alpha(P) \times \omega(P) \geq |X|$*

Erdős–Szekeres theorem (Thm 2.4.6): *An arbitrary sequence $(x_1, \ldots, x_{n^2+1})$ of real numbers contains a monotone sequence (either increasing or decreasing) of length $n+1$*

Def: a *graph* $G = (V, E)$ is such that

- *V is some set (of vertices)*
- *E is a set of 2-subsets of V (called edges)*

Def: the *complete graph* K_n is the graph with n vertices and an edge between every vertex

- *if $\binom{V}{2}$:= all 2-subsets of V, then $E = \binom{V}{2}$*
- $|E_n| = \binom{n}{2} = \frac{n(n-1)}{2} = \sum_{i=1}^{n} i$

Def: the *cycle* C_n is the graph $G = (V, E)$ such that

- $V = \{1, \ldots, n\}$
- $E = \{\{i, i+1\} : i \in \{1, \ldots, n-1\}\} \cup \{n, 1\}$

Def: the *path* P_n is the graph $G = (V, E)$ such that

- $V = \{0, \ldots, n\}$
- $E = \{\{i-1, i\} : i \in \{1, \ldots, n\}\}$

Def: the *complete bipartite graph* $K_{n,m}$ is the graph $G = (V, E)$ such that

- $V = \{u_1, \ldots, u_n\} \cup \{v_1, \ldots, v_m\}$
- $E = \{\{u_i, v_j\} : i \in \{1, \ldots, n\} : j \in \{1, \ldots, m\}\}$

Def (4.1.2): two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be *isomorphic* if there exists a bijection $f : V \to V'$ such that

- $\{x, y\} \in E \Leftrightarrow \{f(x), f(y)\} \in E'$ *holds* $\forall x, y \in V : x \neq y$
- *if such a function exists, we say* f *is an isomorphism*
- *this is denoted* $G \cong G'$

9 Lecture 9

Def (4.2.1): Let G and G' be graphs. We say

- G is a *subgraph* of G' if $V(G) \subseteq V(G')$ and $E(G) \subseteq E(G')$
- G is an *induced subgraph* of G' if $V(G) \subseteq V(G')$ and $E(G) = E(G') \cap \binom{V(G)}{2}$

Def: Let $G = (V, E)$ be a graph. For a given vertex, its *degree*, denoted as $deg_G(v)$ is the number of edges which contain it

Def: We say a subgraph of a graph G is a *path* in G if it is isomorphic to some path P_t

Def: We say a subgraph of a graph G is a *cycle* in G if it is isomorphic to some cycle C_t

10 Lecture 10

Def: Let G be a graph. We say G is *connected* if there exits a path between every two nodes $x, y \in V(G)$

Def: We say $(v_0, e_1, v_1, \ldots, e_t, v_t)$ is a walk of length t from v_0 to v_t if $e_i = \{v_{i-1}, v_i\} \in E \; \forall i \in \{1, \ldots, t\}$

Def: Let $G = (V, E)$ be a graph, and $x, y \in V$. We say $x \sim y$ if there exists a walk from x to y.

- $\sim$ *is an equivalence relation on the vertex set of* G

Def: The *connected components* of a graph are the subgraphs induced by the equivalence classes of $\sim$. Equivalently, the equivalence classes of $\sim$ are all vertices which can be reached by that vertex, which is defined to be a *connected component* of G.

- *The union of all connected components of a graph form the whole graph*

Observation (4.2.2): *Each component of a graph is connected. A graph is connected iff it has a single component*

Def: Let $G = (V, E)$ be a connected graph. Let $v, v' \in V$. The *distance* $d_G(v, v')$ is defined to be the length of the shortest path between v and v'.

- *We can define a distance function* $d : V \times V \to \mathbb{N}$ *by* $(v, v') \mapsto d_G(v, v')$
- $d_G(v, v') \geq 0 : d_G(v, v') = 0 \Leftrightarrow v = v'$
- $\forall v, v' \in V(G) : d_G(v, v') = d_G(v', v)$
- $d_G(v, v'') \leq d_G(v, v') + d_G(v', v'')$
- $d_G(v, v') \in \mathbb{N} : \forall v, v' \in V(G)$
- $d_G(v, v'') \geq 1 \Rightarrow \exists v' \neq v \land v' \neq v'' : d_G(v, v'') = d_G(v, v') + d_G(v', v'')$

Def (4.2.3): Let $G = (V, E)$ be a graph with n vertices, such that $V = \{v_1, \ldots, v_n\}$. The *adjacency matrix* is defined to be the matrix A_G such that

$$a_{ij} = \begin{cases} 1 & : \{v_i, v_j\} \in E \\ 0 & : \{v_i, v_j\} \notin E \end{cases}$$

- *The resultant matrix will be square, and symmetric*

11 Lecture 11

Proposition 4.2.4: Let $G = (V, E)$, and $A = A_G$ be its adjacency matrix. Let A^k denote the *kth* power of the adjacency matrix. Then a^k_{ij} denotes the number of distinct walks from i to j given k steps

Corollary 4.2.5: The distance of any two verticies v_i and v_j satisfies

$$d_G(v_i, v_j) = min(\{k \geq 0 : a^k_{ij} \neq 0\})$$

Def: let $V(G) = \{v_1, \ldots, v_n\}$ be the vertex set of a graph. The *degree sequence* or *graph score* is the set $\{deg(v_1), \ldots, deg(v_n)\}$. Usually, this set will be rearranged in nondecreasing order

Proposition 4.3.1: Let $G = (V, E)$ be a graph. Then

$$\sum_{v \in V} deg_G(v) = 2|E|$$

Corollary 4.3.2: the number of odd-degree vertices in any graph is even

12 Lecture 12

Thm (4.3.3): Let $D = (d_1, \ldots, d_n)$ be a sequence of (nondecreasing) natural numbers, and let $D' = (d'_1, \ldots, d'_{n-1})$ where

$$d'_i = \begin{cases} d_i & : i < n - d_n \\ d_i - 1 & : i \geq n - d_n \end{cases}$$

Then D is a graph score if and only if D' is a graph score

- *apply iteratively until we have reduced the tuple to a manageable length*
- *sort tuple into nondecreasing order after each application*
- *apply corollary 4.3.2 to vet first*

Def: a $k - regular$ graph is a graph such that each vertex is of degree k

Def: an *Eulerian Tour* is a walk such that every vertex is contained and each edge is contained exactly once

Def: a graph is *Eulerian* if it has an Eulerian Tour

Def (4.4.1): a graph G is Eulerian iff

- *G is connected*
- *each vertex has even degree*

13 Lecture 13

Def: a *multigraph* is an ordered pair (V, m) where V is a vertex set and $m : \binom{V}{2} \to \mathbb{N}$ is a function by $m(v_1, v_2) \mapsto \#$ edges between v_1, v_2

Def: a *directed graph* is defined as a pair (V, E) where $E \subseteq V \times V$. Directed edges are defined as ordered pairs $(x, y) \in E$

- *x is referred to as the head, while y is referred to as the tail*

14 Lecture 14

Def: a *directed tour* in a directed graph $G = (V, E)$ is a sequence $(v_0, e_1, \ldots, e_m, v_m)$ such that $e_i = (v_{i-1}, v_i) \in E \ \forall i \in \{0, \ldots, m\}$

Def: a directed graph is *Eulerian* if it has a directed tour such that

- *the directed tour contains all the vertices in V*
- *possesses each directed edge in G (exactly once)*

Def: the *in-degree* of a vertex v in a directed graph is defined as the number of edges ending at v, denoted as $deg_G^+(v)$

Def: the *out-degree* of a vertex v in a directed graph is defined as the number of edges starting at v, denoted as $deg_G^-(v)$

Def: let $G = (V, E)$ be a directed graph. We can apply *symmetrization* to G, producing the graph $sym(G) = (V, \bar{E})$, where $\bar{E} = \{\{x, y\} : (x, y) \in E \vee (y, x) \in E\}$

- *$sym(G)$ is called the symmetrization of G*

Thm (4.5.2): let G be a directed graph. G is Eulerian iff

- *$sym(G)$ is connected*
- $\forall v \in V : deg_G^+(v) = deg_G^-(v)$

15 Lecture 15

Def: a *Hamiltonian path* (or *Hamiltonian circuit*) is a path which passes through each vertex once (except for the endpoints)

- *it need not visit each edge*
- *no easy metric to determine if a graph is hamiltonian*

Def: a graph is *Hamiltonian* if it has a Hamiltonian path

Ore's Thm: let G be an (undirected) graph with n vertices such that $n \geq 3$. Suppose that for any two nonadjacent vertices in $V(G)$, the sum of their degrees is at least n. Then G is Hamiltonian.

- *we cannot say G is not Hamiltonian if it fails to fulfill this condition*

Def: let $G = (V, E)$ be any graph. We say it is

- *k-vertex-connected if it has at least $k + 1$ vertices, and it remains connected if we remove $k - 1$ vertices*
- *k-edge-connected if we get a connected graph when we delete $k - 1$ edges of G*

Def: the *vertex connectivity* of G is the maximum k such that G is k-vertex connected

Def: the *edge connectivity* of G is the maximum k such that G is k-edge connected

Def (4.6.1): let G be a graph. We say it is *2-connected* if

- *it has at least 3 vertices*
- *if, when we delete any single vertex, we (still) have a connected graph*

Def (4.6.2): Let $G = (V, E)$ be a graph. We define:

- *Edge deletion:* $G - e := (V, E \setminus \{e\})$
- *Edge addition:* $G + e := (V, E \cup \{e\})$
- *Vertex deletion:* $G - v := (V \setminus \{v\}, \{e \in E : v \notin e\})$
- *Edge subdivision:* $G\%e := (V \cup \{z\}, (E \setminus \{x, y\}) \cup \{\{x, z\}, \{z, y\}\})$

Thm (4.6.3): a graph is 2 connected iff there exists, for any two vertices of G, a cycle in G containing these two variables

16 Lecture 16

Observation (4.6.4): a graph G is 2-connected iff any subdivision of it is 2-connected

Thm (4.6.5): a graph is 2-connected iff it can be created from a triangle by a sequence of edge subdivisions and edge additions

17 Lecture 17

Def (5.1.1): a *tree* is a connected, acyclic graph

Thm (5.1.2): let $G = (V, E)$ be a graph. The following are equivalent

- *G is a tree*
- *$\forall x, y \in V$ there exists a unique path from x to y in G*
- *the graph G is connected, and deleting any $e \in E$ results in a disconnected graph*
- *the graph G is acyclic, and any graph arising from adding an edge to G contains a cycle*
- *G is connected and $|V| = |E| + 1$*

Def: let $G = (V, E)$ be a graph and $v \in V$. If $deg_G(v) = 1$, we say that v is an *end vertex*, or *leaf node* of G

Lemma (5.1.3): *each tree with at least two vertices contains at least two end vertices*

Lemma (5.1.4): *G is a tree $\Leftrightarrow$ $G - v$ is a tree, where v are the end vertices of G*

18 Lecture 18

Def: we define a *rooted tree* as a pair (T, r)

- *T is a tree*
- *$r \in V(T)$ is a distinguished vertex called the root*

Def: we define a *planted tree* as a rooted tree and a drawing o T, and we mark the root by an arrow

Def: let T and T' be trees. We say $f : V(T) \to V(T')$ is an *isomorphism* of trees T and T' if

- *f is a bijection*
- $\{x, y\} \in E(T) \Leftrightarrow \{f(x), f(y)\} \in E(T')$

If such a map exists we write $T \cong T'$

Def: let (T, r) and (T', r') be rooted trees. We say (T, r) and (T', r') are isomorphic if there exists an isomorphism $f : V(T) \to V(T')$ such that $f(r) = r'$. If such a map exists, we write $(T, r) \cong' (T', r')$

Def: an *isomorphism of planted trees* is an isomorphism of rooted trees such that the isomorphism preserves the left to right ordering of the children of each vertex. If such an isomorphism exists, we write $(T, r, v) \cong'' (T', r', v')$

19 Lecture 19

Isomorphism Algorithm for Planted Trees

K1: Assign 01 to each nonroot end vertex

K2: Let v be the parent of $v_1, \ldots, v_t$. If A_i is the code of the child v_i, the code of vertex v is $0A_1 \ldots A_t 1$

Recovery: Replace 0 in the code with an up arrow, and 1 with a down arrow. Then, write out the code, where up arrows generate a new vertex in the next level, and down arrows take us down to the previous vertex.

Isomorphism Algorithm for Rooted Trees

K1: Assign 01 to each nonroot end vertex

K2: Let v be the parent of $v_1, \ldots, v_t$. If A_i is the code of the child v_i, the code of vertex v is $0A_1 \ldots A_t 1$, where $A_1 \leq \cdots \leq A_t$, where we are using the lexographical ordering

Lexographical ordering for binary strings is defined as followed:

- *A initial segment of $B \Rightarrow A < B$*
- *B initial segment of $A \Rightarrow B < A$*
- **else* let j be the smallest index such that $a_j \neq b_j$*
- *$a_j < b_j \Rightarrow A < B$*
- *$b_j < a_j \Rightarrow B < A$*

Recovery: Replace 0 in the code with an up arrow, and 1 with a down arrow. Then, write out the code, where up arrows generate a new vertex in the next level, and down arrows take us down to the previous vertex.

Def: let $G = (V, E)$ be a graph, with $v \in V$. The *eccentricity* of v, denoted as $ex_G(v)$ is the maximum distance between v and any other vertex of G

Def: let $G = (V, E)$ be a graph. We set $C(G)$, called the *center of* G as the set of vertices of G with minimum eccentricity

Proposition (5.2.1): *For any tree T, $C(T)$ has at most 2 vertices. If it has two vertices v_1 and v_2, then $\{v_1, v_2\}$ is an edge*

Isomorphism Algorithm for Trees

Two trees are isomorphic if and only if they have the same coding. The coding of T is as follows

- *if* $|C(T)| = 1$, *then treat it as a rooted tree with* $r \in C(T)$, *and code appropriately*
- *if* $|C(T)| = 2$, *then consider* $T - e$. *This generates two disconnected rooted trees with codes* A *and* B. *Pick the code with the smaller lexographical ordering*

20 Lecture 24

Dual Graph Definition: let $G = (V, E)$ be a topological planar graph. Let F be the faces of that graph, and the function $\epsilon : E \to F^2$ to map an edge $e \in E$ to $\{F_i, F_j\}$ the pair of faces e delineates. The graph $(F, \epsilon(E))$ denotes the *dual graph* of G

Graph Coloring Greedy Algorithm:
K1: pick a vertex and color it 1
K2: pick an uncolored vertex connected to our colored subgraph and use the lowest valid number to color it
K3: repeat **K2** until all vertices are colored
Note: to color faces, we can convert G to its dual graph, apply coloring, and map back

21 Lecture 25

Proposition (7.1.1): suppose we have a triangular shaped graph partitioned into baby triangles. Label each corner of the big triangle 1, 2, and 3. The edges on the outside of this triangle between end nodes (i, j) may only be labeled i or j. Label the inner nodes whatever we want. Then, at least one baby triangle must have vertex numbering 123. In fact, there must be an odd amount

Proposition (7.1.2): suppose we have a rectangular board whose inside is triangulated. If two players attempt to play a game to traverse opposing diagonal nodes of the game board, a draw is impossible, in the sense that a player will always have a legal move

22 Lecture 26

Proposition (7.1.3): let $f : [0,1] \to [0,1]$ be a continuous function. Then there exists a point $x \in [0,1]$ such that $x = f(x)$

Proposition (7.1.4): every continuous function $f : \Delta \to \Delta$ has a fixed point