

Graph Theory

George C.

March 11, 2023

Contents

0	Fundamental Concepts	5
0.1	Connectivity	5
0.2	Degree Sequences	6
0.3	Bipartite graphs	7
0.4	Walks and Paths	7
0.5	Closed Walks and Cycles	8
0.6	Eulerian Graphs	8
1	Trees	9
1.1	Spanning Trees	10
1.2	Prüfer Codes	10
1.3	Matchings in Bipartite Graphs	11
1.4	Matching Duel Problem	12
2	Connectivity	14
2.1	Connectivity Duel Problem	14
2.2	Max Flow, Min Cut	15
2.3	Ford-Fulkerson	16
3	Graph Coloring	16
3.1	Mycielski Construction	16
3.2	Coloring Extremal Problem	17
3.3	Chromatic Polynomial	18

List of Theorems

1	Definition (graph)	5
	Example (petersen graph)	5
	Note	5
2	Definition (vertex degrees)	5
1	Lemma	5
	Proof	5
3	Definition (complete graph)	5
4	Definition (isomorphic)	5
5	Definition (connected)	5

6	Definition (degree sequence)	6
	Note	6
7	Definition (graphic)	6
	Example	6
1	Theorem (Havel-Hakimi)	6
	Proof	6
	Example	6
8	Definition (bipartite graph)	7
	Example	7
2	Theorem (bipartite iff no odd cycle)	7
	Proof	7
9	Definition (walk)	7
10	Definition (length)	7
11	Definition (path)	7
2	Lemma	8
12	Definition (closed walk)	8
13	Definition (cycle)	8
3	Lemma (closed walk)	8
	Proof	8
	Remark	8
	Remark (local to global results)	8
14	Definition (euler tour)	8
15	Definition (eulerian)	8
1	Proposition (eulerian local property)	8
	Example (seven bridges of Königsberg)	8
	Note	9
16	Definition (tree)	9
	Example	9
17	Definition (leaf)	9
4	Lemma	9
	Proof	9
5	Lemma	9
	Proof	9
18	Definition (spanning tree)	10
6	Lemma	10
	Proof	10
2	Proposition	10
	Proof	10
19	Definition (prüfer code)	10
	Example (prüfer code to tree)	10
3	Theorem	10
	Proof	10
4	Theorem (Cayley)	11
20	Definition (matching)	11
	Example	11
21	Definition (saturates)	11
	Remark	11
3	Proposition (saturated matching for bipartite graph)	11

5	Theorem (Hall's Matching)	11
	Proof	11
	Corollary	12
	Proof	12
22	Definition (maximum)	12
23	Definition (vertex cover)	12
	Example	12
	Remark	12
7	Lemma	13
	Proof	13
	Note	13
6	Theorem (König)	13
	Remark	13
24	Definition (m-augmenting-path)	13
7	Theorem (maximum matching)	13
	Proof	13
25	Definition (stable matching)	13
8	Theorem (Gale-Shapely)	13
	Proof	14
26	Definition (vertex cut)	14
27	Definition (vertex connectivity)	14
8	Lemma	14
	Proof	14
28	Definition (vertex connectivity between two vertices)	14
	Note	14
29	Definition (disjoint path)	15
30	Definition (max pairwise disjoint)	15
9	Theorem (Menger)	15
	Proof	15
31	Definition (network)	15
32	Definition (flow)	15
33	Definition (value)	15
34	Definition (cut)	15
35	Definition (capacity)	15
10	Theorem (Ford-Fulkerson)	16
	Note	16
	Example (exam scheduling)	16
36	Definition (vertex coloring)	16
37	Definition (chromatic number)	16
9	Lemma	16
11	Theorem (Brooks)	16
	Remark	16
	Example (mycielski construction)	16
12	Theorem (Mycielski)	17
	Proof	17
	Remark	17
	Note	17
	Question (coloring extremal problems)	17

38	Definition (turan graphs)	17
	Proof	17
	Example	17
39	Definition (chromatic polynomial)	18
	Example (chromatic polynomial of a complete graph)	18
	Example (chromatic polynomial of a tree)	18
13	Theorem	18
	Proof	18
40	Definition (contraction formula)	18
	Example (sudoku)	18

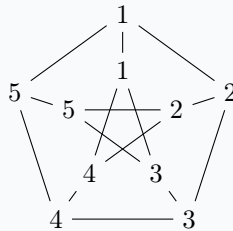
Lecture 1

0 Fundamental Concepts

1/26/2023

Definition 1 (graph). A graph is a pair (V, E) where V is the vertex space and E is the edge space.

Example (petersen graph). An element subset of $\{1, 2, 3, 4, 5\}$ connected by disjointness



Note. In this course we are excluding multiple edges and self loops

Definition 2 (vertex degrees). Let, $G = (V, E)$, $v \in V$, $e \in E$ e are incident if $v \in e$ i.e. v is an endpoint of e

Lemma 1.

$$\sum_{v \in V} \deg(v) = \sum_{v \in V} \sum_{v \in e} 1 = \sum_{v \in e} \sum_{v \in V} 1 = 2|E|$$

Proof. Every edge has two vertices ■

Definition 3 (complete graph). Represented K_n , the graph has $V = 1 \dots n$ and all possible edges

$$|E| = \frac{n(n-1)}{2} = \binom{n}{2}$$

Definition 4 (isomorphic). Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection, $f : v_1 \rightarrow v_2$, s.t. $\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$

0.1 Connectivity

Definition 5 (connected). u and v are connected if there exists a path from u to v . A graph is connected when all vertices are connected

Lecture 2

0.2 Degree Sequences

2023-01-31

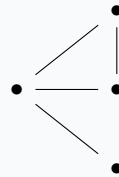
Definition 6 (degree sequence). List of vertex degrees in decreasing order

Note. Isomorphic graphs \Rightarrow same degree sequence; however, if they have the same degree sequence they are not necessarily isomorphic

Definition 7 (graphic). A sequence is graphic if it's the degree sequence of some graph

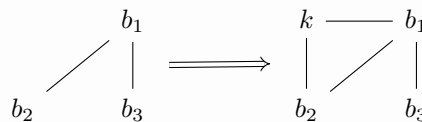
Example. Here are some degree sequences which may or may not exist:

- $(3, 2, 1, 1)$ - not possible since they don't sum to an even number
- $(3, 3, 1, 1)$ - not graphic since there are not enough vertices
- $(3, 2, 2, 1)$ - graphic



Theorem 1 (Havel-Hakimi). $(a_1 \dots a_n)$ is graphic iff $(a_2 - 1, a_3 - 1, \dots, a_{a_1+1} - 1, a_{a_1+2} \dots a_n)$ is graphic

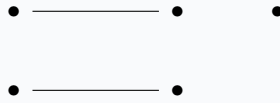
Proof. We can apply the theorem in reverse to understand the intuition. If we have a graphic sequence, $(b_1 \dots b_m)$, we can add a vertex with degree k such that $(k, b_1 + 1, \dots, b_k + 1, b_{k+1} \dots b_m)$.



In the reverse direction, we can subtract a vertex from a graph, and we make a transformation (2-switch) which preserves degree sequences but makes the graph maximal, to get the reverse result. ■

Example. $(3, 3, 3, 3, 3, 2, 2, 1)$ eventually becomes $(1, 1, 1, 1, 0)$ which we can show is

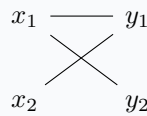
graphic:



0.3 Bipartite graphs

Definition 8 (bipartite graph). A graph is bipartite if it's possible to color vertices using only 2 colors. A simple check for it being bipartite is to check if there are no odd cycles

Example.



Here $G = (X \sqcup Y, E)$ where $x_1, x_2 \in X$ and $y_1, y_2 \in Y$

Theorem 2 (bipartite iff no odd cycle). A graph is bipartite \Leftrightarrow if it contains no odd cycle

Proof. It is clear that if a graph contains an odd cycle that it is not bipartite, so we have to prove the reverse direction: if a graph does not contain an odd cycle then it is bipartite. It suffices to prove the statement for connected graphs - apply the argument to each component.

By induction:

Base Case: When $|E| = 0$ there are no connected vertices so there is no odd cycle and the graph must be bipartite since no vertices are connected to vertices of the same color.

Inductive Step: Assume a graph s.t. $|E| = n - 1$ and no odd cycles is bipartite. Any connected vertices of the same color must be connected by a path of even vertices. Let us add an edge which maintains the restriction that there is no odd cycle in the graph. Suppose an edge is added between these vertices, then the graph is no longer bipartite since two vertices of the same color are connected, the addition of an edge will create an odd cycle — a contradiction. Therefore, when adding an edge to keep the graph bipartite, there must be no odd cycle. ■

0.4 Walks and Paths

Definition 9 (walk). A walk is a sequence of vertices that are connected by edges

Definition 10 (length). The number of edges contained in the walk

Definition 11 (path). A path is a walk that has unique vertices

Lemma 2. A walk from v_0 to v_n implies a path from v_0 to v_n

0.5 Closed Walks and Cycles

Definition 12 (closed walk). A closed walk is a walk which starts and ends at the same vertex

Definition 13 (cycle). A cycle is a closed walk that has unique vertices

Lemma 3 (closed walk). A closed walk of odd length contains an odd cycle

Proof. By induction:

Base Case: $k = 1$ A closed walk of length 3 ($2k + 1$) must be a cycle

Inductive Step: If all vertices in the walk are distinct, we are done since it is a cycle of odd length. In the other case where there are repeated, we can split the walk on repeated vertices to get smaller walks which are proved in the previous cases ■

Lecture 3

Remark. Algorithmic Bipartite Testing

2023-02-02

- Brute force: $2^{|V|} \cdot |E|$
- Proof Algorithm: where we color the vertices and check the edges $|V| + |E|$

Remark (local to global results). Global properties always lead to local results (ex. bipartite implies no odd cycles). In West it is called "TONCAS"

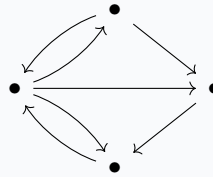
0.6 Eulerian Graphs

Definition 14 (euler tour). A closed walk that visits each edge of a graph exactly once

Definition 15 (eulerian). A graph with an euler tour is Eulerian.

Proposition 1 (eulerian local property). A graph is Eulerian iff every vertex has even degree.

Example (seven bridges of Königsberg).



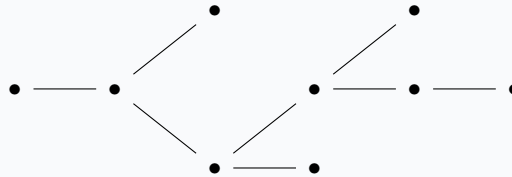
Using a graph to model the seven bridges of Königsberg shows that an Euler tour is not possible.

Note. This is another example of local and global properties where the parity of the degree is the local property and Eulerian is the global property

1 Trees

Definition 16 (tree). A tree is a **connected** and **acyclic** graph

Example.



Definition 17 (leaf). A node of degree 1

Lemma 4. Any connected subgraph of a tree is a tree

Proof. By contradiction:

Assume that the connected subgraph is not a tree, then the subgraph has a cycle, therefore the graph has a cycle, but trees are acyclic. Therefore, contradiction. ■

Lemma 5. A tree with n vertices has $n - 1$ edges

Proof. By induction:

Base Case: There are 0 edges in a 1 vertex tree

Inductive Step: Suppose this holds for n . Let T be a tree with $n + 1$ vertices, then by removing one vertex, which must be a leaf — removing a non-leaf would make the graph unconnected, you remove one edge and the graph becomes the n case. Therefore, the $n + 1$ tree has 1 more edge than an n tree. ■

1.1 Spanning Trees

Definition 18 (spanning tree). A spanning tree (ST) is a subgraph of a **connected** that is a **tree** that contains all vertices of the original graph

Lemma 6. There is a spanning tree in every connected graph

Proof. By contradiction:

Assume G is a connected graph with no ST. Let T be a connected subgraph of G that has the same vertices as G with the smallest number of edges. Since T is not a tree, it does not have a cycle. However, T containing a cycle would imply that T is not a subgraph that has the smallest number of edges. Therefore, contradiction test ■

Proposition 2. With a graph G where $T, T' \subset G$ s.t. $e \in T, e \notin T'$. Then there is edge $e' \in T'$ s.t. $T - e + e'$ is a spanning tree.

Proof. T is a tree $\Rightarrow T - e$ is disconnected. T, T' have the same vertex set $\Rightarrow \exists e \in T'$ that connects two components of $T - e$. Let's claim that $T - e + e'$ is a spanning tree. It has the same number of edges and is connected, so it's a tree, and is spanning since it connects all edges. ■

Lecture 4

1.2 Prüfer Codes

2023-02-09

Definition 19 (prüfer code). A code derived from a tree by iteratively removing the smallest degree vertex and adjoining its neighbor until there are only two vertices left.

Example (prüfer code to tree). Let's take the example $(1, 4, 1, 1)$

We know that vertices of deg 1 do not appear, so we start with an edge from 2 to 1 then we remove 1 from the list. Next we start with 4 and add the next deg 1 vertex. Now we have a connection from 4 to 1 since it's left in the Prüfer code. Then we finally create a tree with vertices 1, 5, 6 and adjoin it.

Theorem 3. For each sequence $(a_1 \dots a_{n-2})$ there exists a unique tree T with $P(T) = (a_1 \dots a_{n-2})$

Proof. By induction:

Base Case: $n = 2$

Sequences have length $n - 2 = 0$ and there is only one tree with 2 vertices

Inductive Step:

For any tree with code $(a_1 \dots a_{n-2})$ Let x be the smallest index not in the sequence then we can construct an edge between it and a_1 and remove a_1 from the code. Therefore, we have a code with one less element. And, by induction, $\exists!$ a tree T' with vertices

$\{1 \dots n\} \setminus x$ and code $(a_2 \dots a_{n-2})$.

The uniqueness comes from the fact that T' is unique and there is only one way to add an edge ■

Theorem 4 (Cayley). There is a bijection between trees and these sequences. Therefore, we can count trees easily by counting sequences.

Therefore, we find there are n^{n-2} trees given n vertices

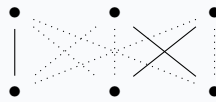
Lecture 5

Matchings

2023-02-14

Definition 20 (matching). A matching is a subset of $M \subset E$ such that no two edges in M share a vertex

Example.



Definition 21 (saturates). For $U \subset V$ say M saturates U if each $u \in U$ is incident to some $e \in M$. A matching that saturates V is called perfect

Remark. Perfect matching relate to bijections.

1.3 Matchings in Bipartite Graphs

For $G = (X \sqcup Y, E)$, is there a matching that saturates X ?

Proposition 3 (saturated matching for bipartite graph). Local to Global: Let $N(S)$ be set of vertices connected to vertices in S . The local property is that for $S \subseteq X$, $|S| \leq |N(S)|$

Theorem 5 (Hall's Matching). $G = (X \sqcup Y, E)$ bipartite has matching saturating X iff $(\forall S \subseteq X)(|S| \leq |N(S)|)$

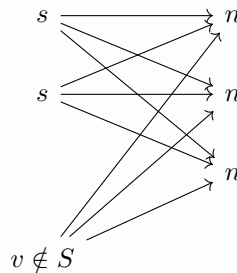
Proof. The proof requires the maximum matching theorem. Suppose G does not have a matching that saturates X . Let M max matching that does not saturate X , $u \in X$ unsaturated. $S = \{x \in X \text{ connected to } u \text{ by a } M\text{-alt path}\}$ $T = \{y \in Y \text{ connected to } u \text{ by a } M\text{-alt path}\}$ Any M -alt path from u to $y \in T$ can be extended uniquely to M -alt path ending in S . Therefore, there is a bijection from T to $S \setminus u$

$N(S) = T$ since $T \subset N(S)$ by definition and $N(S) \subset T$ because we can fix a $y \in N(S)$ which is connected to an x which is connected to u . ■

Corollary. Any k -regular bipartite graph has perfect matching

Proof. Fix $G = (X \cup Y, E)$ bipartite and k -regular

Show $|X| = |Y|$: $k|X| = \sum_{x \in X} \deg(x) = |E| = \sum_{y \in Y} \deg(y) = k|Y|$ Fix $S \in X$. Each edge leaving S lands in $N(S)$. There are $k|S|$ edges leaving S . Therefore, $k|S| \leq k|N(S)|$.

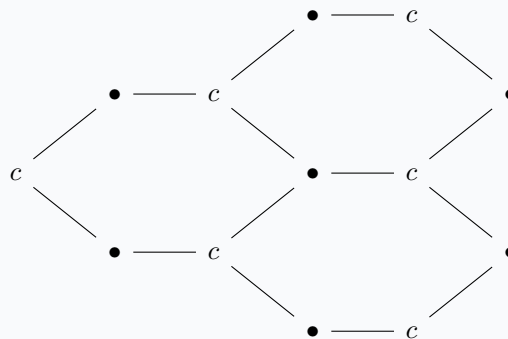


1.4 Matching Dual Problem

Definition 22 (maximum). A matching $M \in E$ is maximum if it has the most edges of any matching

Definition 23 (vertex cover). A vertex cover of G is a subset $Q \subset V$ s.t. every edge is incident to some vertex in Q

Example.



Where vertices labeled c are a vertex cover

Remark. Matchings and vertex covers are dual problems

Lecture 6

Lemma 7. Let Q be a vertex cover, M a matching Then $|M| \leq |Q|$

2023-02-16

Proof. For each $e \in M$ there is at least one vertex of Q incident to e . No two $e, e' \in M$ share a vertex of Q ■

Note. We can use this to show that if we have a vertex cover that covers all vertices with the same size as a matching we must have a maximum matching/minimum vertex cover.

Theorem 6 (König). Let G bipartite, then the max size of the matching is equal to the min size of the vertex cover

Remark. Therefore, matching and vertex covers are dual problems

Definition 24 (m-augmenting-path). A path whose edges alternate between M and $E \setminus M$ whose end points are unsaturated.

Theorem 7 (maximum matching). Local to Global property: If there's no M-augmenting path then M is maximum

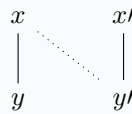
Proof. Suppose M is not a maximum there exists an M' s.t. $|m| < |M'|$ consider the symmetric difference $M \triangle M'$. Here vertices have either degree 1 or 2 and the graph is a union of even cycles and paths.

Let M be a non-maximal matching, M' maximum matching. Consider $M \triangle M'$ By above this is a union of paths and even cycles. Since M and M' share the same size in cycles then some component in $M \triangle M'$ has a path ■

Lecture 7

Definition 25 (stable matching). A matching is stable if

2023-02-23



then either x prefers y to y' or y' prefers x' to x

Theorem 8 (Gale-Shapely). Stable matchings always exist in bipartite graphs

Proof. We perform an algorithm to construct a stable matching:

Round 1:

Each $x \in X$ proposes to top choice

Each $x \in X$ proposes to top choice

Subsequent Rounds: Each unmatched x proposes to top choice not yet proposed to. Each y accepts the best proposal possible breaking an engagement

Repeat

Features

1. Algorithm stops after $< |E|$ rounds
2. End matching is stable
3. Proposers get the best match among all stable matchings. Proposals get worse.

■

Lecture 8

2 Connectivity

2023-02-28

Definition 26 (vertex cut). A vertex cut on a graph G is a subset $S \subset V$ s.t. $G \setminus S$ is disconnected

Definition 27 (vertex connectivity). Represented $\kappa(G)$, the vertex connectivity is the minimum size of a vertex cut of G . For K_n , κ is not defined.

Lemma 8. If G is not complete then G has a vertex cut

Proof. Let G not be complete, then there exist $u, v \in V$ s.t. $\{u, v\} \notin E$. Then $S = V \setminus \{u, v\}$ is a vertex cut ■

2.1 Connectivity Duel Problem

Definition 28 (vertex connectivity between two vertices). Let $G = (V, E)$, for $x, y \in V$ define $\kappa(x, y)$ min size of vertex cut that disconnects x and y

Note. We see that the points we have to remove are related to paths from x to y which are disjoint.

Definition 29 (disjoint path). A path is disjoint with another path if there are no interior vertices they share

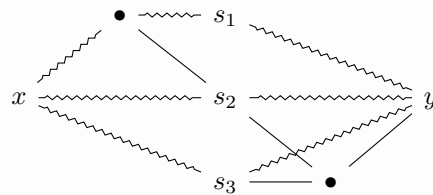
Definition 30 (max pairwise disjoint). Let $\lambda(x, y)$ be the max number of pairwise disjoint x, y -paths

Theorem 9 (Menger). Let $G = (V, E)$, $x, y \in V$, $\{x, y\} \notin E$ then $\kappa(x, y) = \lambda(x, y)$

Proof. $\kappa(x, y) \geq \lambda(x, y)$ is easy
 $\kappa(x, y) \leq \lambda(x, y)$:

Let S be a minimum vertex cut. We want to show that there are $|S|$ disjoint (x, y) paths. Consider two subgroups G_x and G_y where G_x is the union of all paths from x to S (doesn't contain vertices from S) and G_y is the union of all paths from y to S (doesn't contain vertices from S). We have broken up the problem into just working with finding paths to S from x and y .

To form a picture like:



■

2.2 Max Flow, Min Cut

Definition 31 (network). Is a tuple G, s, t, c where $G = (V, E)$ is a directed graph, $s, t \in V$ "source" and "terminus", $c : E \rightarrow \mathbb{N}$ capacity.

Definition 32 (flow). A flow on a network is $f : E \rightarrow \mathbb{N}$ s.t. $f(e) \leq c(e)$ and conservation law $f^+(v) = \sum_{e \rightarrow v} f(e) = \sum_{v \rightarrow e} f(e) = f^-(v)$

Definition 33 (value). Value of f is $f^-(s)$

The general problem is that given a network what is the max value of a flow

Definition 34 (cut). A cut is a partition $V = S \sqcup T$ with $s \in S$, $t \in T$

Definition 35 (capacity). The capacity of a cut S, T is $\sum_e c(e)$ where e is an edge from S to T

2.3 Ford-Fulkerson

Theorem 10 (Ford-Fulkerson). Max value of the flow is equal to the min capacity of a cut

Lecture 9

Note. Capacity of any cut gives upper bound on value of any flow. From this we can see the reasoning of why Ford-Fulkerson finds this as a dual problem

2023-03-02

Lecture 10

3 Graph Coloring

2023-03-07

Example (exam scheduling). Define $G = (V, E)$ where V are courses and E are conflicts if a student takes both courses. Here a graph coloring can help us answer how to schedule classes to limit conflicts.

Definition 36 (vertex coloring). Is a function of $V \rightarrow \mathbb{N}$ s.t. if $\{u, v\} \in E$ then $c(u) \neq c(v)$

Definition 37 (chromatic number). Represented $\chi(G)$, the chromatic number is the minimum number of colors needed to color a graph.

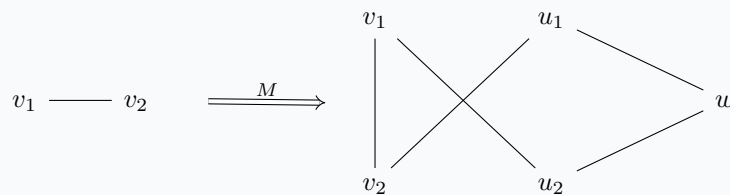
Lemma 9. $\chi(G) \leq \Delta G + 1$ where ΔG is the max degree of a vertex in G

Theorem 11 (Brooks). $\chi(G) = \Delta(G) + 1 \Rightarrow G$ is an odd cycle or complete

Remark. G contains K_n implies $\chi(G) \geq n$, n is called the clique number denoted $\omega(G)$

3.1 Mycielski Construction

Example (mycielski construction).



With construction that if $G = (V, E)$ where $V = \{v_1 \dots v_n\}$ then $M(G)$ has vertices $\{v_1 \dots v_n, u_1 \dots u_n, w\}$ and edges $\{v_i, v_j\}, \{v_i, u_j\}, \{u_i, w\}$, for $\{v_i, v_j\} \in E$

Theorem 12 (Mycielski).

1. $\chi(G) = k \Rightarrow \chi(M(G)) = k + 1$
2. G doesn't contain $K_3 \Rightarrow M(G)$ doesn't contain K_3

Proof. *First statement:* Given k -coloring of G , we can $(k + 1)$ color $M(G)$, where you color u_i the same as v_i and w the $k + 1$ color. We also want to show that the graph has no smaller k coloring. Suppose $M(G)$ has a k -coloring then U uses a $k - 1$ coloring and since the U coloring can be sent to the V coloring. Therefore, G has a $k - 1$ coloring - a contradiction. ■

Remark. Therefore, the clique number is not a very strong property

3.2 Coloring Extremal Problem

Note. Coloring graphs is an NP problem, so it might be better to try to solve extremal problems

Question (coloring extremal problems). Among graphs with a $\chi(G) = k$ what is the maximal/minimal number of edges?

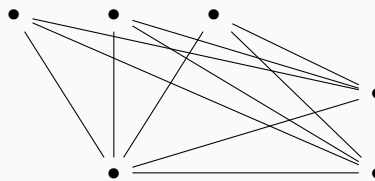
Lecture 11

Definition 38 (turan graphs). Graphs of the form $T_{n,k} = M_{\underbrace{q \dots q}_{k-r}, \underbrace{q+1 \dots q+1}_r}$ are the solutions to the maximal coloring problem.

2023-03-09

Proof. Pretty obvious with calculus based approach ■

Example. $T_{6,3} = M_{3,2,1}$



3.3 Chromatic Polynomial

Definition 39 (chromatic polynomial). $\chi(G, t)$ = number of colorings of G using at most t colors

1. Degree = $|V|$
2. $a_{n-1} = |E|$
3. $\chi(G)$ = smallest t s.t. $\chi(G, t) \neq 0$ ($0, \dots, \chi(G) - 1$ are roots of $\chi(G, t)$)
4. t^d divides $\chi(G, t) \Rightarrow G$ has $\geq d$ components
5. coefficients are log concave

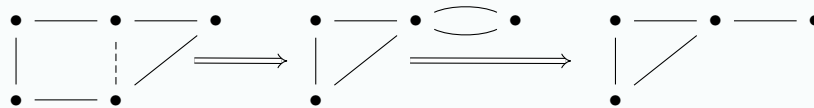
Example (chromatic polynomial of a complete graph). $\chi(K_n, t) = t(t-1)(t-2) \dots (t-(n+1)) = \binom{t}{n}$

Example (chromatic polynomial of a tree). Given a tree T , and a subtree S with one less vertex. Inductively, $\chi(T, t) = (t-1)\chi(S, t)$. And the coloring does not depend on what tree it is, only the number of vertices. $\chi(T, t) = t \cdot (t-1)^{n-1}$

Theorem 13. The chromatic polynomial is a polynomial

Proof. $\chi(G, t) = \chi(G \setminus e, t) + \chi(G \cdot e, t)$
Where $G \cdot e$ is the contraction formula

Definition 40 (contraction formula).



Example (sudoku). We form a graph where vertices are squares on the board and vertices are connected if they are in the same row column or square on the grid. Now this is made into a coloring problem.

How many 9×9 Sudoku puzzles?

Each graph has 81 vertices with degree 20, so there are 810 edges.