

Graph Theory

Vasu Menon

Budapest Semesters in Math
Autumn 2025

Contents

Syllabus	2
Set 1	4
Set 2	8
Set 3	13
Set 4	15
Set 5	19
Set 6	23
Set 7	26
Set 8	33
Set 9	36
Midterm 1	40
Midterm 2	42
Scores	43

Syllabus

Instructor: Kristóf Bérczi¹

The course begins with an overview of the fundamental concepts of graph theory, followed by a detailed discussion of classical topics in both structural and algorithmic graph theory.

Topics

1. **Basics:** Graphs, degrees, paths and cycles, independent sets, cliques, isomorphism, subgraphs, complement of a graph, trees, breadth-first search, depth-first search, applications.
2. **Trees:** Prüfer code, Cayley's theorem, exchange properties of spanning trees, Kruskal's algorithm, Matrix-Tree Theorem.
3. **Eulerian Tours and Hamilton Cycles:** Königsberg bridges problem, Euler's theorem, Hamiltonian cycles and paths, necessary conditions, sufficient conditions (Dirac, Ore, Pósa, Chvátal).
4. **Shortest Paths:** PERT method, Dijkstra's algorithm, Gallai's theorem.
5. **Matchings in Bipartite Graphs:** König-Hall-Frobenius theorem, with König's algorithm, Egerváry's theorem and algorithm, stable matchings, Gale-Shapley algorithm.
6. **Matchings in General Graphs:** Tutte's theorem, matchability of a subset, Petersen's theorem.
7. **Colorings:** Chromatic number and clique number, Mycielski's construction, Brooks' theorem, edge-chromatic number, Vizing's theorem, list colorings.
8. **Flows and Circulations:** Hoffman's theorem, Max Flow-Min Cut Theorem, Menger's theorems, disjoint paths between terminal pairs, nowhere-zero flows, balanced orientations.
9. **Planar Graphs:** Four Color Theorem, Euler's formula, bounding the number of edges of planar graphs, Kuratowski's theorem, graph minors,

¹Department of Mathematics, Eötvös Loránd University, Budapest, Hungary

Wagner's theorem, Five Color Theorem, Schnyder labelings, drawing planar graphs on grids.

10. **Higher Connectivity:** Highly edge- and node-connected graphs, block decompositions, ear decompositions.
11. **Orientations:** Degree bounds, Hakimi's lemma, Luigi's conjecture, F-avoiding orientations.
12. **Partially Ordered Sets:** Dilworth's theorem.
13. **Perfect Graphs:** Basic examples, perfectness of comparability graphs, Perfect Graph Theorem, Strong Perfect Graph Theorem.

Set 1

Problem 1

Decide if there exists a graph with the following degree sequence:

$$1, 1, 1, 2, 4, 5, 6, 6, 7$$

Solution. There does not exist such a graph. The sum of the degrees of a graph's vertices must be even ($2|E|$), but here, the sum of the degrees is 33, which is odd.

Problem 2

Prove that any simple graph on at least 2 vertices contains two vertices having the same degree.

Proof. Suppose a simple graph $G = (V, E)$ has $n \geq 2$ vertices. Then for any $v \in V$, $d(v) \leq n - 1$ and $d(v) \in \{0, \dots, n - 1\}$. Since we have n vertices and $n - 1$ degree "options", at least two vertices must fall in the same degree "option". Note that if $d(v) = 0$ for some $v \in V$, then you cannot have another vertex $v' \in V$ with $d(v') = n - 1$. Thus, there are at least two vertices having the same degree. \square

Problem 3

Verify that for any graph $G = (V, E)$, either G or its compliment \overline{G} is connected.

Proof. If G is connected, then we are done. Otherwise, if G is not connected, then there exist vertices $u, v \in V$ such that there is no path from u to v . This implies that there are two subgraphs $A = (V_A, E_A)$ and $B = (V_B, E_B)$ of the graph $G = (V, E)$ such that $V_A \cap V_B = \emptyset$ and $V_A \cup V_B = V$ and for every $a \in V_A$ and $b \in V_B$, $ab \notin E$. Now, in the compliment $\overline{G} = (\overline{V}, \overline{E})$, for every $a \in V_A$ and $b \in V_B$, $ab \in \overline{E}$. Thus there exists paths between any vertex in V_A and any vertex in V_B , and any vertices $u, v \in V_B$ has a path via a vertex in V_A and any vertices $u, v \in V_A$ has a path via a vertex in V_B . So \overline{G} is connected. \square

Problem 4

Prove that if $G = (V, E)$ is a simple graph on $2n$ vertices such that every vertex has degree at least n , then G is connected.

Proof. We will proceed inductively. For the base case, assume $n = 1$. A graph with 2 vertices with a connection between them is connected. Assume a simple graph on $2n$ vertices such that every vertex has degree at least $n > 1$ is connected. We will prove that a graph on $2(n+1) = 2n+2$ vertices such that every vertex has at least degree $n+1$ is connected as well. We are taking the connected graph with $2n$ vertices and adding 2 additional vertices. We are also then taking each vertex and adding 1 to its degree, effectively adding an edge that it previously did not have. Note that these 2 new vertices can either connect to itself, or connect to a vertex that exists within this connected section. Since these vertices each has degree at least $n+1$, each of them must connect back to the connected path of the graph. Thus, this newly constructed graph is connected as well. \square

Problem 5

Prove that any connected graph has a vertex whose deletion (together with the edges incident to it) results in a connected graph.

Proof. We know that the vertices of a connected graph G can always be enumerated as v_1, \dots, v_n so that $G_i := G[v_1, \dots, v_i]$ is connected for every i . Setting $i = n$, we have $G_n = G[v_1, \dots, v_n]$. Consider the vertex v_n . Deleting it results in the graph $G_{n-1} = G[v_1, \dots, v_{n-1}]$ which is still connected. \square

Problem 6

Prove that if $G = (V, E)$ is a graph with at least 5 vertices, then either G or \overline{G} contains a cycle.

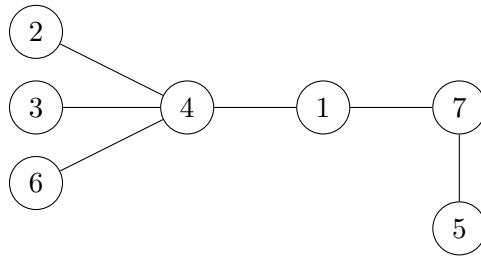
Proof. If G contains a cycle, then we are done. Otherwise, let G be an n vertex graph without a cycle with $n > 5$. Then G is a forest with $1 \leq d \leq 5$ trees. We know then that the number of edges $|E_G| \leq n - 1$. In \overline{G} , however,

we have $|E_{\bar{G}}| = |E_{K_n}| - |E_G| = \frac{n(n-1)}{2} - |E_G| \geq \frac{n(n-1)}{2} + n - 1$ ². Note then that for all $n > 5$, $|E_{\bar{G}}| \geq \frac{n(n-1)}{2} + n - 1 > n - 1$. Since this compliment graph has greater than $n - 1$ edges, it has more edges than it's spanning tree, and thus is no longer a tree and contains a cycle. \square

Problem 7

Determine the tree whose Prüfer code is $(4, 4, 7, 4, 1)$.

Solution.



Problem 8

Characterize those trees whose Prüfer code consists of all different numbers.

Solution. If a tree with length n has a Prüfer code has no repeating number, then the tree is just a straight path with length n . The code has length $n - 2$. In the tree, there are 2 vertices with degree 1, and they don't appear in the code. The other $n - 2$ vertices each appear once in the code, which means they have degree 2, and these are the inner vertices of the path.

Problem 9

Let T be a tree whose Prüfer code consists of the same number. What is the maximum length of a path in T ?

Solution. If the Prüfer code of T consists of the same number, then there is exactly one vertex (whose label is the only number in the Prüfer code) with

²We know $|E_{K_n}| = \frac{n(n-1)}{2}$ because given n vertices, we first connect one of them to the other $n - 1$ vertices, then another one to the remaining $n - 2$ vertices and so on. In the end we have the sum $\sum_1^{n-1} = \frac{n(n-1)}{2}$.

many leaves. So the maximum length of a path in T is 2, with this maximal path going from a leaf to the main node to another leaf.

Problem 10

Determine the number of trees on vertices $1, \dots, n$ in which vertex 1 has degree exactly 2.

Solution. We can count the number of Prüfer codes (with length $n - 2$) that represent this tree. Since vertex 1 has degree exactly 2, it appears exactly once in the code, and there are $n - 2$ choices where it can go. We then have $n - 1$ other labels with $n - 3$ positions where they can go. Thus, we have $(n - 2)(n - 1)^{n - 3}$ such trees.

Set 2

Problem 1

Let $G = (V, E)$ be a simple, connected graph with adjacency matrix A

- (a) Prove that, for any positive integer k , the (i, j) -th entry of A^k equals the number of walks of length k from vertex i to vertex j in G .
- (b) Use this property to show that the number of walks of length 2 starting and ending at vertex i is equal to the degree of i .
- (c) Determine the graph G if every entry of the matrix $A + A^2$ is equal.

Proof. Suppose $G = (V, E)$ is a simple, connected graph with adjacency matrix A .

a) Note that for vertices i and j , since the (i, j) -th entry of A corresponds to the number of edges between i and j , it equivalently represents the number of walks of length 1 (an edge) that exists between i and j . Additionally, we observe that

$$(A^k)_{ij} = \sum_{v_1, v_2, \dots, v_{k-1} \in V} A_{iv_1} A_{v_1 v_2} A_{v_2 v_3} \cdots A_{v_{k-1} j}$$

for any positive integer k where $i, v_1, \dots, v_{k-1}, j \in V$ are a sequence of vertices. Because we sum over all intermediary vertices $v_1, v_2, \dots, v_{k-1} \in V$, the (i, j) -th entry of A^k corresponds to the number of ways to get from vertex i to vertex j through vertices v_1, v_2, \dots, v_{k-1} , which is exactly the number of walks of length k from vertex i to vertex j in G .

b) The number of walks of length 2 starting and ending at vertex i is equivalent to $(A^2)_{ii}$. Then, observe that

$$(A^2)_{ii} = A_{i1}A_{1i} + A_{i2}A_{2i} + \cdots + A_{in}A_{ni} = \sum_{j=1}^n A_{ij}A_{ji}$$

Note that each term $A_{ij}A_{ji}$ equals 1 only when i and j are adjacent, and 0 otherwise. The summed quantity thus accumulates the number of edges that are connected to vertex i , which is exactly the degree of i .

c) Observe that $(A + A^2)_{ij}$ counts the number of walks of length 1 or 2 that exist between vertices i and j . Since all entries are constant, then the

number of walks that are either length 1 or 2 between any two vertices are the same. Note that since the diagonal entries of A are all 0 and the diagonal entries of $A + A^2$ are equal, then the diagonal entries of A^2 must equal the non-diagonal entries of $A + A^2$. Thus, the number of paths of length 2 from any vertex i back to itself must be equal to the number of paths of length 1 or 2 between any two distinct vertices. This is only possible if every pair of distinct vertices is adjacent. Thus, the graph must be complete. Note that the trivial solution $A = 0_{n \times n}$ also satisfies the condition. Therefore, the graph G must be either a set of vertices with no edges such that $G = (V, \emptyset)$, or the complete graph $G = K_n$. \square

Problem 2

Prove that the edge set of a graph $G = (V, E)$ can be partitioned into two spanning trees if and only if G can be constructed from a single vertex using the following two operations:

- (i) adding a new vertex v together with two edges connecting v to existing vertices,
- (ii) subdividing an existing edge uw by introducing a new vertex v , replacing uw with uv and vw , and connecting v to an existing vertex with an additional edge.

Proof. We will prove both directions.

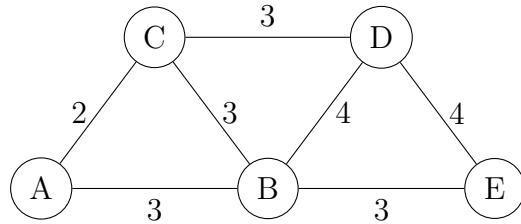
(\implies) Assume G is constructed from the two operations. We need to show it can be partitioned into 2 spanning trees. We will proceed by induction. For the base case, assume G has no edges yet. Here, it can be partitioned into 2 spanning trees, each spanning tree consisting of 0 edges. Now, assume G can be partitioned into 2 spanning trees T_1 and T_2 . If we apply operation 1, we add a vertex v to G connected by two separate edges e_1 and e_2 . T_1 and T_2 can receive e_1 and e_2 respectively and T_1 and T_2 remain trees. Thus, G remains being able to be partitioned into 2 spanning trees. If instead we apply operation 2, then we consider some edge uw , where uw is some edge either in T_1 or T_2 . This edge is then replaced with uv and vw , connecting v to an existing vertex with an additional edge. Without loss of generality, suppose uw is a member of T_2 . The replacement edges uv and vw then will belong to the same tree T_2 , allowing u , v , and w to remain connected in T_2 . The final edge placed to connect the new vertex v with any other vertex will

then always return to a connected segment of the other tree, T_1 , without creating a cycle and thus preserving the partitions. Therefore, any graph G constructed from the two operations can be partitioned into 2 spanning trees.

(\Leftarrow) We will reverse the above argument. \square

Problem 3

Consider the following graph. Run Kruskal's algorithm step by step to find its minimum-weight spanning tree. At each step, indicate which edge is added and the current total weight of the partial spanning forest.



Solution.

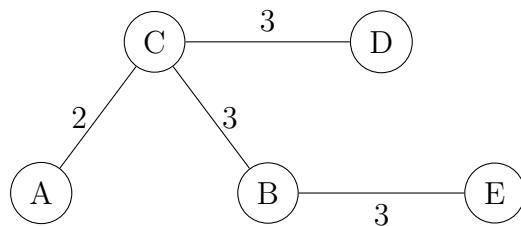
Step 1: Add minimum edge AC with cost 2. Current total cost = 2.

Step 2: Add next minimum edge CB with cost 3. Current total cost = 5.

Step 3: Add next minimum edge CD with cost 3. Current total cost = 8.

Step 4: Add next minimum edge BE with cost 3. Current total cost = 11.

All vertices are now connected and we have avoided cycles. The following is our spanning tree:



Problem 4

Let $G = (V, E)$ be a connected graph with distinct edge weights. Prove that the minimum-weight spanning tree of G is unique

Proof. Assume for the sake of contradiction that we have two minimum-weight spanning trees of G , A and B , where $A \neq B$. Without loss of generality, this implies that there exists a set of edges $E = A \setminus B$ with $E \neq \emptyset$. We will then apply the exchange theorem. Let $e \in E$ to be the edge has minimum weight. If we add e to B , then we will have formed a cycle. Additionally, there is at least one edge f in this cycle such that $f \in B \setminus A$. Note that since all edge weights of the graph G are distinct, all the edges in this cycle are strictly greater than the weight of the edge e , and thus, the weight of f is strictly greater than the weight of e . Remove f from this newly constructed B yields a spanning tree B' . Note that the total weight of $B' < B$, which implies that B is not minimal, a contradiction. Therefore, $A = B$ and the minimum-weight spanning tree of G is unique. \square

Problem 5

Suppose we stop Kruskal's algorithm as soon as we have selected $k - 1$ edges, where k is a fixed integer between 1 and $|V|$. Show that the resulting forest is optimal in the sense that it minimizes the total weight among all spanning forests with k connected components.

Proof. TODO \square

Problem 6

Prove that every connected graph admits a closed walk that uses each edge exactly twice.

Proof. Let G be a connected graph. Construct a new graph G' by duplicating and doubling every edge in G . Then every vertex of G' has even degree, and G' is connected. Since every vertex has an even degree, we know that G has a Eulerian tour, which is a closed walk that traverses each edge in G' exactly once. Since each edge in G appears twice in G' , this walk corresponds to a closed walk in G that traverses each edge in G exactly twice. \square

Problem 7

Let G be a graph in which the degree of every vertex is 4. (A graph with all degrees equal to k is called *k-regular*, so G is a 4-regular graph.) Prove that we can color all edges of G with one of the two colors red and blue, such that every vertex becomes incident to exactly two red and two blue edges.

Proof. Since every vertex has degree 4, all vertices have even degree, and thus there exists a Eulerian tour through every edge. Consider a starting vertex v and begin a Eulerian tour through the graph, coloring each red and blue alternating between the two. Now every vertex has 2 colored incident edges, 1 red and 1 blue. If we ignore or remove all these edges from the graph, we are left with a graph where each vertex has degree 2. Note that here, every vertex still has an even degree, and thus, there exists another Eulerian tour through every edge. If we begin another Eulerian tour, coloring each visited red and blue alternating between them, we will have colored all edges, and thus, we are left with a graph where every vertex becomes incident to exactly two red and two blue edges. \square

Problem 8

Let G be a graph in which every vertex has even degree. Show that it is possible to label the edges of G with the integers $1, \dots, m$ (where $m = |E(G)|$) so that, for every vertex, the greatest common divisor of the labels on its incident edges is 1.

Proof. Since G is a graph in which every vertex has even degree, there exists a Eulerian path. Choose an arbitrary starting vertex v and begin a Eulerian tour through every edge. For the first edge, label it 1. Then label the next edge 2, and continue this pattern. That is, for each subsequent edge, label it with an integer that is 1 greater than the previous edge. Observe that through this process, we have labeled the edges of G with the integers $1, \dots, m$ where $m = |E(G)|$. Additionally, note that for every edge, there exists an edge with some integer p and another integer $p + 1$. The greatest common divisor of the labels is therefore 1, since for any $p \in \mathbb{Z}_{>0}$, $\gcd(p, p + 1) = 1$. \square

Set 3

Problem 1

Prove that Pósa's theorem implies Ore's theorem

Proof. For a graph $G = (V, E)$, assume Ore's theorem's condition, that is, assume $d(u) + d(v) \geq n$ for every $u, v \in V$ with $uv \notin E$. Since $uv \notin E$, then $d(u) \leq n - 2$, since u cannot be connected to itself and the vertex v . Similarly, $d(v) \leq n - 2$. Thus, $d(u) + d(v) \leq 2n - 4$, and since $d(u) \geq 1$ and $d(v) \geq 1$, we have that $d(u) + d(v) \geq 2$. This implies that $2 \leq d(u) + d(v) \leq 2n - 4 \implies 2 \leq 2n - 4 \implies n \geq 3$. Now, suppose $1 \leq d_1 \leq d_2 \leq \dots \leq d_n < n$ is the degree sequence. Then, for all $i < j \leq \frac{n}{2}$, $i + j < n$, and since we know that by Ore's condition, $d_i + d_j \geq n$, we have that $i < j \leq i + j < n \leq d_i + d_j$, and thus, $i < j < d_i + d_j$. This implies that for all $d_i \geq i + 1$, which is exactly Pósa's condition, and thus, there exists a Hamilton cycle. Therefore, Pósa's theorem implies Ore's theorem. \square

Problem 2

Let G be a graph on $2x + 1$ vertices with all degrees at least x . Prove that G contains a Hamilton path.

Proof. Suppose G is a graph with $n = 2x + 1$ vertices and all vertices have degree at least x . Now, add a vertex v_0 and connect it to all $2x + 1$ vertices, and thus, $d(v_0) = 2x + 1$. Consequently, we have $n = 2x + 2$ vertices such that for every vertex $v \in V$, $d(v) \geq x + 1 = \frac{2x+2}{2} = \frac{n}{2}$. Thus, by Dirac's theorem, we have a Hamilton cycle. Removing the vertex v_0 from the graph would then leave it without a Hamilton cycle, but a Hamilton path would exist. \square

Problem 3

Let G be a regular simple graph on 2025 vertices that contains a Hamilton cycle. Prove that G also contains a Eulerian circuit. (Reminder: We say that a graph is regular if all of its vertices have equal degree.)

Proof. Suppose G is a regular simple graph with 2025 vertices that contains a Hamilton cycle. To prove that G also contains an Eulerian circuit, we need

to show that G is connected and that every vertex in G has an even degree. Since G contains a Hamilton cycle, then we know it must be connected. Furthermore, since the graph is regular, suppose each vertex has degree d . Then, for 2025 vertices, the sum of the degrees, $\sum d(v) = 2025d$. By the Handshake Lemma, we know that this sum must be even, and thus, d must be even. Therefore, G also contains a Eulerian circuit. \square

Problem 4

An oriented complete graph is called a tournament. Show that every tournament contains a directed Hamilton path.

Proof. We will proceed by induction. For the base case, consider a tournament with 1 vertex. Trivially, there exists a directed Hamilton path. Now for the inductive step, suppose the tournament with n vertices contains a directed Hamilton path enumerated as v_1, v_2, \dots, v_n . Consider adding a vertex u and ensuring that all other vertices have a directed edge connecting it. If we have all other vertices outwardly pointing to u , then there exists the Hamilton path v_1, v_2, \dots, v_n, u . If u has outwardly directed edges to all other vertices, then there exists the Hamilton path u, v_1, v_2, \dots, v_n . Consider the first vertex in the sequence v_1, v_2, \dots, v_n that u has an outward edge pointing to. Call this vertex v_i . Note that then, v_{i-1} points to u , and thus, there exists the Hamilton path $v_1, v_2, \dots, v_{i-1}, u, v_i, \dots, v_n$. Therefore, every tournament contains a directed Hamilton path. \square

Problem 5

Suppose that a graph G can be partitioned into k edge-disjoint spanning trees. Let s be a specified node of G . Prove that G can be partitioned into k edge-disjoint spanning trees T_1, \dots, T_k in such a way that the trees are equitable at s in the sense that $|d_{T_i}(s) - d_{T_j}(s)| \leq 1$ for every $i, j \in [k]$.

Proof. TODO \square

Set 4

Problem 1

Let $D = (V, A)$ be a directed graph and let $c : A \rightarrow \mathbb{R}$ be a conservative weight function. Is it true that there always exists a nonnegative feasible potential?

Proof. Yes. Let $n = |V|$ and let $s \in V$ be some vertex. Suppose $\pi(v) := \min(\text{cost of a walk from } s \text{ to } v \text{ with at most } n - 1 \text{ edges})$. Consider an edge $uv \in A$ and take the minimum walk from s to u with at most $n - 1$ edges (which has cost $\pi(u)$) and add the edge uv to the walk. Now we have a walk from s to v with a cost $\pi(u) + c(uv)$ and therefore, $\pi(v) \leq \pi(u) + c(uv)$. Rearranging yields $\pi(v) - \pi(u) \leq c(uv)$ and thus, π is a feasible potential. Now, let $\pi'(v) := \pi(v) - m$ where m is the minimum feasible potential among all vertices. Here, we have that $\pi'(v) \geq 0$ for all $v \in V$ and is thus a nonnegative feasible potential, and therefore, there always exists a nonnegative feasible potential. \square

Problem 2

Let $D = (V, A)$ be a directed graph and let $c : A \rightarrow \mathbb{R}$ be a conservative weight function. Is it true that any subpath of a cheapest $s-t$ path is itself a cheapest path between its endpoints?

Proof. Yes. Let $\mu(s-t)$ be the minimum cost of an $s-t$ path. Let $u \in V$ be a vertex contained inside of this path. Here, $s-u$ is an arbitrary subpath of the $s-t$ path. Then, since $C(s-t) = \mu(s-t)$, we have that $\mu(s-t) = C(s-u) + C(u-t)$ where C denotes the cost of a path. Note that since c is conservative, there are no negative cycles, and C is therefore well-defined. If $C(s-u)$ was not the minimum cost of an $s-u$, then $\mu(s-t)$ would no longer be minimal, thus, $C(s-u) = \mu(s-u)$. Therefore, $\mu(s-u)$ is the cost of the cheapest $s-u$ path, which is some subpath of the $s-t$ path. \square

Problem 3

Let $D = (V, A)$ be a directed graph and let $c : A \rightarrow \mathbb{Z}$ be a conservative weight function. Suppose π_1 and π_2 are both feasible potentials with respect to c .

- (a) Prove that $\lfloor(\pi_1 + \pi_2)/2\rfloor$ and $\lceil(\pi_1 + \pi_2)/2\rceil$ are also feasible potentials.
- (b) Prove that $\max(\pi_1, \pi_2)$ and $\min(\pi_1, \pi_2)$ are feasible potentials.

Proof. Assume $\pi_1(v) - \pi_1(u) \leq c(uv)$ and $\pi_2(v) - \pi_2(u) \leq c(uv)$ for all edges $uv \in A$

- (a) Define $\pi_{avg} := \frac{\pi_1 + \pi_2}{2}$. Then, summing up the two inequalities in our assumption and dividing by 2 yields $\frac{\pi_1(v) + \pi_2(v)}{2} - \frac{\pi_1(u) + \pi_2(u)}{2} \leq c(uv) \implies \pi_{avg}(v) - \pi_{avg}(u) \leq c(uv)$, and thus, π_{avg} is feasible. Now, since the costs are integers, we have that $\lfloor \pi_{avg}(v) \rfloor - \lfloor \pi_{avg}(u) \rfloor \leq \pi_{avg}(v) - \pi_{avg}(u) \leq c(uv)$ and that $\lceil \pi_{avg}(v) \rceil - \lceil \pi_{avg}(u) \rceil \leq \pi_{avg}(v) - \pi_{avg}(u) \leq c(uv)$. Thus, $\lfloor(\pi_1 + \pi_2)/2\rfloor$ and $\lceil(\pi_1 + \pi_2)/2\rceil$ are feasible potentials.
- (b) Define $\pi_{max}(v) := \max(\pi_1(v), \pi_2(v))$. Note that for any edge $uv \in A$, $\pi_{max} \leq \max(\pi_1(u) + c(uv), \pi_2(u) + c(u, v))$. Since $c(uv)$ is a constant, we have that $\max(\pi_1(u) + c(uv), \pi_2(u) + c(u, v)) = \pi_{max}(u) + c(uv)$. This implies that $\pi_{max}(v) \leq \pi_{max}(u) + c(u, v) \implies \pi_{max}(v) - \pi_{max}(u) \leq c(u, v)$, and thus, π_{max} is a feasible potential. We will proceed similarly for $\pi_{min} := \min(\pi_1(v), \pi_2(v))$. Observe that $\pi_{min}(v) \leq \min(\pi_1(u) + c(uv), \pi_2(u) + c(uv)) \implies \min(\pi_1(u) + c(uv), \pi_2(u) + c(uv)) = \pi_{min}(u) + c(uv)$. So, $\pi_{min}(v) \leq \pi_{min}(u) + c(uv)$ and therefore, $\pi_{min}(v) - \pi_{min}(u) \leq c(uv)$. Thus, π_{min} is a feasible potential as well.

□

Problem 4

Let $D = (V, A)$ be a directed graph and let $c : A \rightarrow \mathbb{Z}$ be a conservative weight function. Furthermore, let P be a cheapest s - t path. Reverse the edges of P and negate the costs of these edges. Show that the resulting cost function is also conservative in the graph thus obtained.

Proof. TODO

□

Problem 5

Develop an algorithm to decide whether a directed graph with a conservative weight function contains a zero-weight cycle.

Proof. TODO

□

Problem 6

Given a directed graph with a conservative weight function on the edges and a lower and upper bound specified for each vertex, how can one decide whether there exists a feasible potential within the given bounds?

Proof. We need to find way to detect if there exists a π such that $\pi(v) - \pi(u) \leq c(u, v)$ and $l(v) \leq \pi(v) \leq u(v)$ for all $v \in V$. To do so, apply the following process:

Consider a vertex v_0 with $\pi(v_0) = 0$. Then, $\pi(v) \leq u(v) \implies \pi(v) - \pi(v_0) \leq u(v)$. Moreover, $\pi(v) \geq l(v) \implies \pi(v_0) - \pi(v) \leq -l(v)$. Consider these differences as directed edges on a graph, since they are of the same form as the feasible potential constraint $\pi(v) - \pi(u) \leq c(u, v)$. Represent the inequality $\pi(v) - \pi(v_0) \leq u(v)$ as an edge from v_0 to v with cost $u(v)$, and represent the inequality $\pi(v_0) - \pi(v) \leq -l(v)$ as an edge from v to v_0 with cost $-l(v)$. Note that in this graph, if there exists a negative cycle, then π is unfeasible and thus, there does not exist a feasible potential within the given bounds. Therefore, we must now detect the existence of negative cycles precisely by using the Bellman-Ford algorithm. If Bellman-Ford detects a negative cycle, no feasible potential exists, otherwise, there is a feasible potential within the given bounds.

□

Problem 7

Let P be a path, and let \mathcal{F} be a family of its subpaths. Develop an algorithm that select a collection of edge-disjoint subpaths from \mathcal{F} so that the total length of the selected subpaths is maximized.

Solution. The algorithm will run as follows. First, take \mathcal{F} and sort it so that each subpath is ordered based on the position of its ending vertex in P where the smallest ending vertex is first. Then, initialize an empty list and iterate through every subpath s . Add s to a list if s does not share any

edges with subpaths already in the list, otherwise, skip it. The collection of subpaths in the list will be the selection of edge-disjoint subpaths from \mathcal{F} such that the total length of the selected subpaths is maximized.

Problem 8

Let $D = (V, A)$ be a directed graph and let $c : A \rightarrow \mathbb{Z}$ be a conservative weight function. Recall that $\pi_c^n(v)$ denotes the minimum cost of a walk of length at most n ending at v . Prove that among all nonpositive feasible potentials, $\pi_c^n(v)$ is the largest; that is, for any feasible potential π , we have $\pi(v) \leq \pi_c^n(v)$ for all $v \in V$.

Proof. Let π be a nonpositive feasible potential. Let $W = (v_0, v_1, \dots, v_k)$ with $v_k = v$ and $k \leq n$ denote a walk of length at most n ending at v . We know that $\pi(v_i) - \pi(v_{i+1}) \leq c(v_i, v_{i+1})$ for all i . Then, $\pi(v) - \pi(v_0) \leq \sum_{i=0}^{k-1} c(v_i, v_{i+1}) = C(W)$. Since π is nonpositive, we have that $\pi(v_0) \leq 0$ and thus $\pi(v) \leq C(W)$. Since this holds for all walks of length at most n ending at v , it also holds for the minimum cost walk, and thus, $\pi(v) \leq \pi_c^n(v)$ for all $v \in V$. \square

Set 5

Problem 1

Let $G = (S, T; E)$ be a bipartite graph. Suppose that G has a matching covering $X \subseteq S$ and another matching covering $Y \subseteq T$. Prove that G contains a matching that covers every vertex in $X \cup Y$.

Proof. Denote the matching covering $X \subseteq S$ by $M_S \subseteq E$ and the matching covering $Y \subseteq T$ by $M_T \subseteq E$. Consider $M = M_S \cup M_T$. Note that every vertex is either an isolated vertex or has degree 1 or 2, in which case it is part some even cycle or path. Consequently, we have 3 distinct components, either an even cycle C , a path P , or isolated vertices. We can safely ignore all isolated vertices since isolated vertices are incident to no edges and is thus not covered by M_S nor M_T , implying that is is not an element of $X \cup Y$. Consider the component C , which contains edges that alternate between M_S and M_T . Without loss of generality, choose M_T and take all the $M_T \cap C$. Here, all vertices in C are covered by this choice of edges. Now consider the component P . If both endpoints of P are in X , choose $M_S \cap P$. Here, these chosen edges alternate in the path P but still cover $X \cap P$. If both endpoints of P are in Y , choose $M_T \cap P$ by the same logic. In this way, we have essentially chosen a matching $M = M_S \cup M_T$ that covers $X \cup Y$, and thus, G contains a matching that covers every vertex in $X \cup Y$. \square

Problem 2

Let $G = (S, T; E)$ be a bipartite graph, and for each $v \in S \cup T$, let \prec_v be a linear order on the edges incident to v . Let $e = uv$ be the edge most preferred by u with respect to \prec_u . Furthermore, assume that $e \prec_v f$ for some edge f . Prove that the sets of stable matchings of $G - f$ and G coincide.

Proof. We will show that the sets of stable matchings of $G - f$ and G coincide, that is, $f \notin M$ for any stable matching M of G . Let M be a stable matching of G . Assume $f \in M$ for the sake of contradiction. Then, we have $f = vw$ for some vertex w . Clearly, $e \notin M$. Since both u and v prefer e over their current matching edge, e is a blocking edge, and consequently, M is not a stable matching, a contradiction. Thus, $f \notin M$, implying that the sets of stable matchings of $G - f$ and G coincide. \square

Problem 3

The outcome of the Gale-Shapley (proposal) algorithm is boy-optimal, meaning that each boy receives the best possible partner he can obtain in any stable matching, and girl-pessimal, meaning that each girl is matched to the worst possible partner she can obtain in any stable matching.

Proof. TODO

□

Problem 4

Show that any two stable matchings cover the same set of vertices.

Proof. Suppose M_1 and M_2 are two matchings that do not cover the same vertices. So there exists some vertex v that is matched in M_1 but not in M_2 . Let v be matched to the vertex u and so $uv \in M_1$. Since v is unmatched in M_2 , $uv \notin M_2$ and thus u is matched in M_2 to some other vertex w . Note that u cannot be unmatched in M_2 since that would lead to uv being unmatched edge with both endpoints unmatched in M_2 , contradicting stability. Consider u being matched to $w \neq v$. Since $uv \in M_1$, u prefers v over being unmatched and maybe even over w . If u prefers v over w , then uv is a blocking edge in M_2 . If u prefers w over v , then v is matched to u but may prefer another option in M_1 , but in M_2 , since v is unmatched, v would prefer u over being unmatched. Either way, uv is a blocking edge in M_2 , and thus M_2 matching is not stable. Therefore, no vertex can be matched in M_1 and unmatched in M_2 . Symmetrically, no vertex can be matched in M_2 and unmatched in M_1 . So, the set of matched matched vertices is identical in M_1 and M_2 and thus, any two stable matchings cover the same set of vertices. □

Problem 5

If M_1 and M_2 are stable matchings in the bipartite graph G , and each boy chooses from $M_1 \cup M_2$ the edge he prefers more, then we obtain a stable matching $M_1 \vee M_2$, in which each girl is matched via the less preferred edge from $M_1 \cup M_2$. If, on the other hand, the girls choose their preferred edge, then we obtain stable matching $M_1 \wedge M_2$, in which each boy is matched via the less preferred edge from $M_1 \cup M_2$.

Proof. TODO

□

Problem 6

Let $G = (V, E)$ be a (not necessarily bipartite) graph, and for each $v \in V$, let \prec_v be a linear order on the edges incident to v . A matching $M \subseteq E$ is called stable if there is no edge $e = uv \in E \setminus M$ such that both u and v prefer e to their respective edges in M . Prove that if G is not bipartite, then there exists an assignment of preference lists to the vertices of G for which no stable matching exists.

Proof. If G is not bipartite, it must contain a cycle of odd length. Denote this cycle by $C = (v_0, \dots, v_{n-1})$ where n is an odd integer. For any vertex v_i , place the vertex appearing directly to its clockwise direction v_{i+1} first and the vertex behind it v_{i-1} second in the preference lists. Populate the rest of the preference list arbitrarily. Assume for the sake of contradiction that there exists a stable matching M . Because two adjacent edges cannot be in M and the cycle is odd, then there must exist two adjacent edges that do not belong in M . In other words, there are two edges in the cycle $e_j = v_j v_{j+1}$ and $e_{j+1} = v_{j+1} v_{j+2}$ which are both not in M . Here, the vertex v_{j+1} is not matched to neither its direct clockwise vertex nor its direct counterclockwise vertex and is therefore matched with some other vertex further down its preference list. The vertex v_j prefers the edge e_j connecting it to v_{j+1} but since $e_j \notin M$, it is connected elsewhere. We then have two vertices v_j and v_{j+1} that prefer the edge e_j over their current match which implies the matching is not stable, a contradiction. \square

Problem 7

Let k, n be positive integers satisfying $k < n/2$. Prove that for every k -element subset A of $[n] = \{1, \dots, n\}$, one can add an element from $[n] \setminus A$ to A in such a way that the resulting $\binom{n}{k}$ $(k+1)$ -element subsets of $[n]$ are all different.

Proof. Let S be the collection of all k -element subsets of $[n]$ and let T be the collection of all $(k+1)$ -element subsets of $[n]$. We will use S and T to construct a disjoint graph G . The edges between S and T represents the “adding an element” process in our statement. Consider $A \in S$ to be a k -element subset and $B \in T$ to be a $(k+1)$ -element subsets. Add an edge between A and B if and only if $A \subset B$. Note that the degree of A is $n - k$ and the degree of B is $k + 1$. Now consider an arbitrary subset $X \subseteq S$ and

let $\Gamma(X)$ denote the neighborhood of X in G . There are $|X|(n - k)$ total edges incident to Z and $|\Gamma(X)|(k + 1)$ total edges incident $\Gamma(X)$. We then have $|X|(n - k) \leq |\Gamma(X)|(k + 1)$ and thus $\frac{|X|}{|\Gamma(X)|} \leq \frac{k+1}{n-k}$. Moreover, since we have that $k < n/2$, $n - k > k + 1$ and thus $\frac{k+1}{n-k} \leq 1$. Consequently, $\frac{|X|}{|\Gamma(X)|} \leq 1$, implying $|X| \leq |\Gamma(X)|$. By our Corollary³, this implies that there exists an S matching. Therefore, for every k -element subset of $[n]$, there exists a $(k + 1)$ -element subset that is the result of appending a separate element from $[n]$. Thus, we have proven that for every k -element subset A of $[n] = \{1, \dots, n\}$, one can add an element from $[n] \setminus A$ to A in such a way that the resulting $\binom{n}{k}$ $(k + 1)$ -element subsets of $[n]$ are all different. \square

³Given $G = (S, T; E)$, there exists a matching covering S if and only if for all $X \subseteq S$, $|\Gamma(X)| \geq |X|$

Set 6

Throughout, let $D = (V, A)$ be a digraph and $s, t \in V$.

Problem 1

Let $c: A \rightarrow \mathbb{R}_+$ be a capacity function, and let x and x' be maximum flows. Prove that if there exists an augmenting path from s to some vertex v in D_x , then there also exists one in $D_{x'}$.

Proof. First, since x and x' are both maximum flows, the MFMC theorem implies that they carry their maximum capacities through all minimum cuts. Let $(S, V \setminus S)$ be the set of vertices reachable from s in D_x with $v \in S$. Now consider the minimum cut $(S, V \setminus S)$. Because x' is also a maximum flow, it must carry the maximum flow through this same cut. That is, for any edge (u, w) with $u \in S$ and $w \in V \setminus S$, we have $x'(u, w) = c(u, w)$. Consequently, in the graph $D_{x'}$, no forward edge crosses from S to $V \setminus S$, but backward edges may exist. Now, for any vertex $u \in S$, there exists a path from s to u in D_x . Since x' also maximizes flow across the same cut, the capacities within S (edges between vertices in S) remain identical to x . Therefore, the same path from s to v that exists in D_x exists in $D_{x'}$ restricted to S . Thus, there exists an augmenting path from s to v in $D_{x'}$ as well.

□

Problem 2

Suppose that, in addition to the edge capacities $c: A \rightarrow \mathbb{R}_+$, each vertex is assigned an upper bound specifying the maximum amount of flow that may pass through it. Design an algorithm that computes a maximum flow subject to these vertex capacity constraints.

Solution. Our algorithm will augment D to construct a new $D' = (V', A')$ in which we can run a known algorithm on. For each vertex $v \in V$, split it into two vertices a_v, b_v and add a directed edge (a_v, b_v) from a_v to b_v with the capacity $m(v)$, which will denote its maximum flow upper bound. To preserve the original edges and their capacities, for every $(u, v) \in A$ with capacity $c(u, v)$, add the edge (b_u, a_v) with the same capacity $c(u, v)$. These changes construct a new V' and A' . We will then run the Edmond-Karp

algorithm⁴ which will then compute the maximum flow subject to the vertex capacity constraints. This ensures that any flow through v in D corresponds to a flow from a_v to b_v in D' , which is upper-bounded by $m(v)$.

Problem 3

Let c_1, \dots, c_k be capacity functions on the edges. Give an algorithm for deciding whether there exists an s - t cut S that is simultaneously a minimum cut for every g_i .

Solution. TODO (what is g_i ?)

Problem 4

Given a capacity function $c: A \rightarrow \mathbb{R}_+$, design an algorithm that decides whether the minimum s - t cut is unique.

Solution. First, compute a maximum s - t flow x and construct the graph D_x . Now, let $A = v \in V : v$ is reachable from s in D_x . Note that now the cut $(A, V \setminus A)$ is minimum, since in D_x , the edge (u, v) is present if we can send more flow along it, and if $v \in V \setminus A$, then v is not reachable from s in D_x , and all edges from A to $V \setminus A$ are at maximum capacity with all edges going the other way having zero flow, implying that the total flow across the cut is maximal, and by the MFMC theorem, it is then a minimum cut.

Next, let $B = v \in V : t$ is reachable from v in D_x . For any minimum cut $(S, V \setminus S)$, observe that every vertex in A must be in S , since vertices reachable from s in D_x cannot be on the T side of a minimum cut, and that no vertex in B can be in S , since vertices that can reach t in D_x must lie on the T side of every minimum cut because sending them to S would create a non-maximal edge crossing the cut, which contradicts the minimality of the cut. So for any minimum cut S , we have that $A \subseteq S \subseteq V \setminus B$.

Consider the case $A \neq V \setminus B$, so there exist vertices outside $A \cup B$, which can be assigned either to S or T without changing the capacity of the cut, implying that the minimum cut is not unique. If $A = V \setminus B$, then there are no vertices outside $A \cup B$, so the only way to satisfy $A \subseteq S \subseteq V \setminus B$ is when $S = A$, implying that the minimum cut is unique. This means that

⁴Section 3.5 in Frank's book

the minimum cut $(A, V \setminus A)$ is unique if and only if every vertex belongs to $A \cup B = V$.

Now we will construct an algorithm to create the sets A and B . The algorithm is as follows:

1. Compute a maximum flow x (using Edmonds-Karp).
2. Construct the residual graph D_x .
3. Run BFS/DFS from s in D_x to find A .
4. Run BFS/DFS from t in the reversed D_x to find B .
5. If $A \cup B = V$ (equivalently $A = V \setminus B$), output that the minimum cut is unique; otherwise, it is not unique.

Problem 5

Give an algorithm that finds all edges for which any positive increase in capacity results in an increase in the value of the maximum flow. Does such an edge always exist? How can these edges be identified?

Solution. TODO (use D_x)

Set 7

A connected graph $G = (V, E)$ is *2-edge-connected* if $G - e$ remains connected for any $e \in E$, and *2-vertex-connected* if $|V| \geq 3$ and $G - v$ remains connected for any $v \in V$. An *ear-decomposition* of G is a sequence of subgraphs (P_0, P_1, \dots, P_k) such that:

1. P_0 is a cycle in G ;
2. for each $i \geq 1$, P_i is a path or a cycle in G such that
 - if P_i is a path, then its end vertices lie in $\bigcup_{j < i} P_j$, but its internal vertices are not contained in $\bigcup_{j < i} P_j$;
 - if P_i is a cycle, then it meets $\bigcup_{j < i} P_j$ in exactly one vertex, and its other vertices are not contained in $\bigcup_{j < i} P_j$.
3. $G = \bigcup_{i=0}^k P_i$.

Problem 1

Prove that if P_0, P_1, \dots, P_k is an ear-decomposition of a graph $G = (V, E)$, then $|E| = |V| + k$.

Proof. We will proceed inductively. Assume P_0 is the ear-decomposition of G . Note that $k = 0$. Then G is entirely a cycle $C = v_1, e_1, v_2, e_2, \dots, v_n, e_n, v_1$. Here, $|E| = n$ and $|V| = n$, and thus, $|E| = |V| + k$ holds true. Now assume that P_0, P_1, \dots, P_{k-1} is the ear-decomposition of G and $|E_{k-1}| = |V_{k-1}| + (k-1)$. Add another subgraph (ear) P_k with q edges. If P_k is a cycle, then it meets the graph at exactly one vertex and so it adds its $q-1$ internal vertices. Else, if P_k is a path, then it meets the graph at two vertices (both of the endpoints), and thus it also introduces $q-1$ vertices. Now, we have that $|E_k| = |E_{k-1}| + q$ and $|V_k| = |V_{k-1}| + (q-1)$. Rearranging gives $|E_{k-1}| = |E_k| - q$ and $|V_{k-1}| = |V_k| - (q-1)$. Substituting these values in $|E_{k-1}| = |V_{k-1}| + (k-1)$ yields $|E_k| - q = |V_k| - (q-1) + (k-1)$. Simplifying gives $|E_k| = |V_k| + k$. Thus, by induction, if P_0, P_1, \dots, P_k is an ear-decomposition of a graph $G = (V, E)$, then $|E| = |V| + k$. \square

Problem 2

Let G be a connected undirected graph. Prove the following equivalences.

- (a) G is 2-edge-connected if and only if it has an ear-decomposition
- (b) G is 2-vertex-connected if and only if it has an open ear-decomposition with $|P_0| \geq 3$.
- (c) G is factor-critical if and only if it has an odd ear-decomposition

Proof. The following are the proofs for the three equivalences.

(a) We will prove both directions.

(\Rightarrow) Assume G is 2-edge-connected. We will construct an ear-decomposition. Since G is 2-edge-connected, the minimum degree is at least 2, so G contains a cycle. Let C_0 be this cycle. Suppose we have constructed a subgraph C_k which is an ear-decomposition. If $C_k = G$, we are done. If $V(C_k) \subset V(G)$ or $E(C_k) \subset E(G)$, since G is connected, there exists an edge $e \in E(G) \setminus E(C_k)$ with at least one endpoint in C_k . If both endpoints of e are in C_k , then e itself forms an ear (a path of length 1). If $e = uv$ with $u \in C_k, v \notin C_k$, then because G is 2-edge-connected, the edge e is not a bridge in G . Therefore, $G - e$ is connected, which implies that there is a path from v to C_k in $G - e$. Let P be a shortest such path. Then $P \cup \{e\}$ forms a path starting and ending in C_k with all internal vertices not in C_k , which is a valid ear. We can then add this ear to form C_{k+1} . Finally, repeat this until G is fully reconstructed.

(\Leftarrow) Assume G has an ear-decomposition $G = P_0, P_1, \dots, P_k$. We will now proceed inductively on the number of ears k . We know that P_0 is a cycle, which is clearly 2-edge-connected. Now assume that P_i is 2-edge-connected. Let $P_{i+1} = P_i \cup Q$, where Q is an ear that is attached to the vertices $u, v \in V(P_i)$. Suppose now that we remove an edge e from P_{i+1} . Then, if $e \in Q$, then P_i is unaffected and is still connected. The removal of e from Q disconnects the path Q , but the vertices of Q are still connected to P_i through the remaining parts of the path attached to u or v . Thus, $P_{i+1} - e$ is connected. If $e \in P_i$, then since P_i is 2-edge-connected, $P_i - e$ remains connected. The path Q remains attached to $P_i - e$ at u and v , so the whole graph remains connected. Thus, G is 2-edge-connected.

(b) We will prove both directions.

\Rightarrow) Assume G is 2-vertex-connected.

\Leftarrow) Assume G has an open ear-decomposition with $|P_0| \geq 3$. We will proceed inductively on the number of ears k . We know that P_0 is a cycle of length ≥ 3 , and this is clearly 2-vertex-connected. Now, assume that P_i is 2-vertex-connected. We create P_{i+1} by adding the open ear Q with distinct endpoints $u, v \in V(P_i)$. Let w be any vertex in P_{i+1} . We need to show that $P_{i+1} - w$ is connected. If w is an “internal” vertex of the new ear Q , then $Q - w$ breaks into two components, but both are attached to P_i . Since P_i is unaffected and still connected, the entire graph remains connected. Now, if $w \in V(P_i)$, then $P_i - w$ is connected by our inductive hypothesis. Q has endpoints u, v , and even if $w = u$, the ear is still connected to P_i at v (since u and v are distinct), and so the vertices of Q stay connected to the graph.

(c) We will prove both directions.

\Rightarrow) Assume G is factor-critical.

\Leftarrow) Assume G has an odd ear-decomposition. We will proceed inductively. P_0 is an odd cycle C_{2k+1} . Then, removing any vertex v leaves a path Q_{2k} , which has a perfect matching (since we can take alternate edges and add it to the matching). Thus, P_0 is factor-critical. Now, assume that P_i is factor-critical. We create P_{i+1} by adding Q where Q is an odd ear. We need to show that for any $x \in V(P_{i+1})$, $P_{i+1} - x$ has a perfect matching. We can proceed with case work. For the first case, suppose $x \in V(P_i)$. Since P_i is factor-critical, $P_i - x$ has a perfect matching M . The ear Q has an odd number of edges, and thus has an even number of internal vertices. The path Q , excluding the endpoints, consists of disjoint edges covering all internal vertices. Let this matching be M_Q . Then $M \cup M_Q$ is a perfect matching of $P_{i+1} - x$. For the second case, assume x is an internal vertex of Q . Here, removing x splits Q into two paths, Q_1 , which is attached to u , and Q_2 , which is attached to v , where u, v are distinct endpoints of the path. Since the total edges in Q was odd, one of these paths has an odd length (with even vertices), and the other has even length (with odd vertices). We can match all vertices in Q_1 internally except the endpoint u and we can match all the vertices in Q_2 similarly. Now, we need to match u inside P_i . Since P_i

is factor-critical, $P_i - u$ has a perfect matching M' . Combing these will then give a perfect matching for $P_{i+1} - x$.

□

Problem 3

- (a) Prove that a connected graph $G = (V, E)$ is 2-edge-connected if and only if for every edge $e \in E$ there exists a cycle in G containing e .
- (b) Prove that a connected graph G is 2-vertex-connected if and only if for every pair of edges $e, f \in E$ there exists a cycle in G containing both e and f .

Proof. Firstly, consider the following claim:

Claim. *In a 2-vertex-connected graph G , for any cycle C and any vertex $z \notin V(C)$, there exist two internally vertex-disjoint paths from z to C whose endpoints on C are distinct.*

Proof of Claim. If every $z - C$ path met C at the same vertex v , then removing v would separate z from the rest of the graph, contradicting 2-vertex-connectivity.

- (a) We will prove both directions.

(\Rightarrow) Assume G is 2-edge-connected. Then, for any edge e , $G - e$ is still connected. This means that e is not a bridge (cut-edge). Note that an edge is a bridge when it does not lie on any cycle, and thus, every edge lies on some cycle.

(\Leftarrow) Suppose that for every edge $e \in E$, there exists a cycle in G containing e . Let $e = uv$ be an arbitrary edge. Since e lies on a cycle C , there is a path in C from u to v that does not use e . Thus even after removing e there remains a $u-v$ path, and so $G - e$ is still connected. Since this holds for every e , G is 2-edge-connected.

- (b) We will prove both directions.

(\Rightarrow) Assume G is 2-vertex-connected. Take two edges e and f . By the same argument in part (a) every edge lies on some cycle. So, pick a cycle C that contains e . If f is on C , then we are done. Else, let $f = xy$ be an edge not on C . Apply our above claim to the endpoints x and y relative to C . Thus, we have that

- 1) There exist two internally disjoint paths from x to C meeting C at some vertex p
- 2) There exist two internally disjoint paths from y to C meeting C at some vertex q .

If we can choose those paths so that the attachment vertices p and q on C are distinct, then we can take the edge of C that is between p and q that contains the edge e (one of the two edges between p and q along the cycle must contain e). Concatenating that edge with the two paths from x and y to p and q and the edge xy creates a cycle that contains both e and f .

Now we need to eliminate the possibility that every choice of x - C and y - C paths meet C only at the same vertex r . If that were the case, removing r would then separate x (and y) from the cycle $C \setminus \{r\}$, so r would be a cut-vertex of G , contradicting our 2-vertex-connectivity property. Thus, we can choose paths with distinct endpoints on C , and the argument above completes the proof that some cycle contains both e and f .

(\Leftarrow) Suppose that for every pair of edges e, f there is a cycle containing both. We will show G has no cut-vertex. Suppose, for the sake of contradiction, that v is a cut-vertex. Then, there are at least two components in $G - v$. Now, pick two distinct components A and B and then pick edges e and f with e entirely inside $A \cup \{v\}$ and f entirely inside $B \cup \{v\}$. This is possible because each component has at least one edge unless it is a single isolated vertex. If a component is a single vertex, then pick the other component that has an edge. Any cycle that contains both e and f would have to then contain v and also include a path from A to B avoiding v , which is impossible because v separates A and B . This contradiction shows no cut-vertex exists, so G is 2-vertex-connected.

□

Problem 4

Let $G = (V, E)$ be a 2-edge-connected 3-regular graph. Prove that G admits a perfect matching.

Proof. Let

$$D(G) = \{v \in V \mid \exists \text{ a maximum matching } M \text{ s.t. } v \text{ is not covered by } M\}$$

$$A(G) = \Gamma(D(G)) \setminus D(G)$$

Now, for the sake of contradiction, assume G does not have a perfect matching. Since there are vertices that must be left out of any matching, we know that $D(G)$ is not empty. Now, let $D = D(G)$ be the set of vertices missed by at least one maximum matching and let $A = N(D) \setminus D$ be the neighbors of D in the rest of the graph. Then let K_1, K_2, \dots, K_q be the connected components of the subgraph formed by the vertex set D . By Gallai's Lemma, each of these components must then be factor-critical. This means that $|V(K_i)|$ is odd, since removing 1 vertex produces a perfect matching, and there are no edges between D and the rest of the graph except for those going to A . Now, we can count the number of edges m connecting the set D and the set A . For any component $K_i \subseteq D$, since G is 3-regular, we have that $\sum_{v \in K_i} \deg(v) = 3|K_i|$. Moreover, since $|K_i|$ is odd, this sum is odd. Additionally, The sum of degrees is also equal to $2 \cdot (\text{edges inside } K_i) + (\text{edges leaving } K_i)$. The sum being odd implies that the number of edges leaving K_i must then be odd. We know that since G is 2-edge-connected (implying there are no bridge edges, see problem 2), the number of edges leaving K_i cannot then be 1. Thus, there are at least 3 edges leaving every component K_i . If we let q be the number of connected components in the set $D(G)$ and m be the number of edges connecting the set D to the set A , we then have that $m \geq 3q$. Furthermore, these m edges must connect to vertices in A . Since G is 3-regular, the vertices in A can take at most 3 edges each, and so we have that $m \leq 3|A|$. Combining the bounds, we have that $3q \leq m \leq 3|A|$, which implies that $q \leq |A|$. However, we know that from the definition of maximum matchings that if we delete A , the graph breaks into the components K_i . A maximum matching must match vertices in A to vertices in distinct components of D , which leaves at least $q - |A|$ vertices unmatched. Since we originally assumed G has no perfect matching, we have that $q - |A| > 0$ and thus, $q > |A|$, which is a contradiction. Thus, our assumption was false, and G must therefore have a perfect matching. \square

Problem 5

Let G be a graph on n vertices with minimum degree at least $n/2$. Show that if n is even, G contains a perfect matching.

Proof. Note that if $n = 2$, we have $G = K_2$, which trivially has a perfect matching. So now suppose that $n \geq 3$. Since we have that for all n vertices $v \in G$, $d(v) \geq \frac{n}{2}$, we satisfy Dirac's condition, which then implies that there exists a Hamiltonian cycle C in G . Let the vertices of this cycle in order be v_1, v_2, \dots, v_{2k} . We can proceed through this cycle and select every alternating edge to construct the matching $M = \{(v_1, v_2), (v_3, v_4), \dots, (v_{2k-1}, v_{2k})\}$. Since the cycle covers all vertices exactly once, the set M hits every vertex exactly once. Thus, M is a perfect matching. \square

Set 8

Problem 1

Determine the chromatic number of the complement of a cycle on n vertices.

Proof. Let C_n be the cycle on n vertices. We know that a proper coloring of the complement graph $\overline{C_n}$ assigns different colors to vertices that are adjacent in $\overline{C_n}$. Vertices are adjacent in $\overline{C_n}$ if and only if they are not adjacent in C_n . Therefore, a set of vertices can share the same color in $\overline{C_n}$ if and only if they are “pairwise non-adjacent” in $\overline{C_n}$, which means they must be “pairwise adjacent” (or the same vertex) in C_n . So a valid color class in $\overline{C_n}$ corresponds to a complete subgraph (clique) in C_n . The only cliques in a cycle C_n ($n \geq 4$) are single vertices (K_1) and edges (K_2). There are no cliques of size 3 or larger. Thus, coloring $\overline{C_n}$ is equivalent to partitioning the vertex set V into disjoint edges and vertices. To minimize the number of colors (cliques), we must maximize the number of vertices paired up into edges. Let k be the number of edges (K_2) used in the partition. These edges must be disjoint, so k cannot exceed the matching number $\nu(C_n) = \lfloor \frac{n}{2} \rfloor$. The remaining $n - 2k$ vertices must be colored individually (as K_1 s). The total number of colors used is $\chi(\overline{C_n}) = k + (n - 2k) = n - k$. To minimize this quantity, we must maximize k . We choose a maximum matching, so $k = \lfloor \frac{n}{2} \rfloor$, and thus the chromatic number $\chi(\overline{C_n}) = n - \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil$. For $n = 1$, we have that $\overline{C_1}$ is a single vertex, so the chromatic number is $1 = \lceil \frac{1}{2} \rceil$. For $n = 2$, we have that $\overline{C_2}$ consists of 2 vertices with no edge between them, so the chromatic number is $1 = \lceil \frac{2}{2} \rceil$. For $n = 3$, we have C_3 , which is a triangle K_3 , and the complement $\overline{C_3}$ is the empty graph on 3 vertices ($3K_1$), where chromatic number is $\chi(\overline{C_3}) = 1 \neq \lceil \frac{3}{2} \rceil$, which is the only exception. Thus, the chromatic number of the complement of a cycle on n vertices is $\lceil \frac{n}{2} \rceil$ for all $n \neq 3$, and when $n = 3$, the chromatic number is 1. \square

Problem 2

Let G be a bipartite graph. Prove that for its complementary graph \overline{G} we have $\chi(\overline{G}) = \omega(\overline{G})$.

Proof. Let G be a bipartite graph with n vertices. First, observe that a clique in \overline{G} corresponds to an independent set in G . Thus, the “clique number” (the size of the largest clique) of the complement is the independence number of

the original graph, $\omega(\overline{G}) = \alpha(G)$. Next, consider the chromatic number $\chi(\overline{G})$. As shown in the proof for Problem 1, a valid coloring of \overline{G} partitions the vertices of G into cliques of G . Since G is bipartite, it contains no cliques of size 3 or greater. Thus, we are partitioning the vertices into disjoint edges and single vertices. If we select a matching M of size $|M|$ in G , we can use the edges of M as $|M|$ color classes. The remaining $n - 2|M|$ vertices must be colored individually. The total number of colors is $|M| + (n - 2|M|) = n - |M|$. To find the chromatic number, we need to minimize this sum, which is the same as maximizing the matching size $|M|$. Thus, we use a maximum matching $\nu(G)$, and so $\chi(\overline{G}) = n - \nu(G)$. In any bipartite graph, the size of the maximum independent set plus the size of the maximum matching equals the total number of vertices, so $\alpha(G) + \nu(G) = n \implies \alpha(G) = n - \nu(G)$. Combining our results yields $\chi(\overline{G}) = n - \nu(G) = \alpha(G) = \omega(\overline{G})$. \square

Problem 3

Draw some lines in the plane so that no three of them go through the same point. Consider their points of intersection as the vertices of a graph and the segments between neighbouring intersection points on each line as edges. Prove that the chromatic number of the resulting graph is at most 3.

Proof. Let the graph created from the process be $G = (V, E)$. The vertices V are the intersection points of the lines. We know that two vertices are connected by an edge if they are consecutive intersection points along one of the lines. Assume that no lines are vertical (since we can rotate the plane a tiny amount if needed). This allows us to order the vertices by their x -coordinate. We will show that every subgraph has a vertex of degree at most 2, which would imply that the graph is 3-colorable. Suppose we have the vertex ordering v_1, v_2, \dots, v_n based on increasing x -coordinates. Any vertex v_i is the intersection of exactly two lines, say L_a and L_b . Looking to the right of v_i (vertices with higher index), the line L_a continues to its next intersection point, and L_b continues to its next intersection point. So v_i is connected to at most two vertices with an index higher than i . This ordering proves that G is 2-degenerate. A coloring algorithm using the reverse of this order (coloring from v_n down to v_1) ensures that when we color v_i , it has at most 2 neighbors that have already been colored (its “future” neighbors in the x -ordering). Since there are 3 available colors, there is always a color free for v_i . Thus, $\chi(G) \leq 3$. \square

Problem 4

For a positive integer t , let $f(t)$ denote the minimum number k for which there exists a graph G with $\chi(G) = t$ and $|E(G)| = k$. Determine the value of $f(t)$ for all positive integers t .

Proof. We want the minimum number of edges in a graph with chromatic number t . Consider a t -critical graph, that is, a graph G with $\chi(G) = t$ such that removing any edge decreases the chromatic number. We know that in a t -critical graph, every vertex has degree at least $t - 1$. Thus, for n vertices, we have that $|E(G)| \geq \frac{n(t-1)}{2}$. Now, to minimize the number of edges, take the smallest number of vertices possible. A t -chromatic graph must have at least t vertices, and the complete graph K_t achieves $\chi(K_t) = t$ with $|E(K_t)| = \frac{t(t-1)}{2}$. Thus, for all positive integers t , $f(t) = \frac{t(t-1)}{2}$ \square

Problem 5

Let G be a simple graph on n vertices and \overline{G} be its complementary graph. Prove that $\chi(G)\chi(\overline{G}) \geq n$.

Proof. Let $\chi(G) = a$ and $\chi(\overline{G}) = b$. Color G with a colors and \overline{G} with b colors. Here, each vertex is assigned an ordered pair (i, j) where $i \in \{1, \dots, a\}$ is its G -color and $j \in \{1, \dots, b\}$ its \overline{G} -color. Note that two distinct vertices cannot share the same pair, since if they did, they would be in the same color class in both G and \overline{G} , so they would be nonadjacent in both G and \overline{G} , which is impossible. Therefore, each of the n vertices receives a distinct pair, giving $ab \geq n$. Thus, $\chi(G)\chi(\overline{G}) \geq n$. \square

Set 9

Problem 1

Prove that if G is a graph on 11 vertices then at least one of G and its complementary graph \overline{G} is not planar.

Proof. We proceed by contradiction. Assume that both G and its complement \overline{G} are planar graphs. Let $n = 11$ be the number of vertices and let m be the number of edges in G . The number of edges in the complement \overline{G} is the total number of possible edges in K_{11} minus m , so $|E(\overline{G})| = \binom{11}{2} - m = 55 - m$. Now, note that for a graph with $n \geq 3$ vertices to be planar, we need that $|E| \leq 3n - 6$. Since we assumed that G is planar, we have that $m \leq 3(11) - 6 = 27$, and since we assumed \overline{G} is planar, we have that $55 - m \leq 3(11) - 6 = 27$. Now, summing these two inequalities yields $m + (55 - m) \leq 27 + 27 \implies 55 \leq 54$, which is a contradiction. Therefore, the assumption that both graphs are planar must be false, and so at least one of them must be non-planar. Thus, if G is a graph on 11 vertices then at least one of G and its complementary graph \overline{G} is not planar. \square

Problem 2

What is the minimum number of edge crossings with which the graph $K_{4,4}$ can be drawn on the plane?

Proof. Let c be the crossing number of $K_{4,4}$. If we remove c edges from the graph (one edge from each crossing), we get a planar subgraph G' . The graph $K_{4,4}$ is a bipartite graph with $n = 8$ vertices and $m = 16$ edges. The planar subgraph G' will have $m' = 16 - c$ edges. Now, for any bipartite planar graph with $n \geq 3$, the number of edges satisfies $|E| \leq 2n - 4$. If we apply this to G' , we have that $16 - c \leq 2(8) - 4 \implies c \geq 4$, and so the crossing number is at least 4. Thus, the minimum number of edge crossings with which the graph $K_{4,4}$ can be drawn on the plane is 4. \square

Problem 3

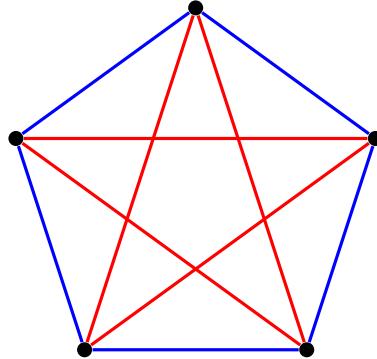
Determine the edge-chromatic number of the d -dimensional hypercube Q_d , where the vertices of Q_d correspond to the elements of $\{0, 1\}^d$ and two vertices are connected if their Hamming distance is 1.

Proof. We will first discuss some properties of the graph Q_d . Since vertices correspond to the elements of $\{0, 1\}^d$, we have that $|V(Q_d)| = 2^d$, and $|E(Q_d)| = d(2^{d-1})$. We will proceed by induction on d . For the base case where $d = 1$, the graph Q_1 is a single edge, which requires 1 color. Thus, $\chi'(Q_1) = 1$. Now for the inductive step, assume $\chi'(Q_d) = d$. The graph Q_{d+1} consists of two copies of Q_d . By the inductive hypothesis, we can color the edges within both copies using d colors. The edges that connect the two copies join each vertex in the first copy to its corresponding vertex in the second. Since these connecting edges are a perfect matching, we can assign all of them a single new color, $d + 1$. Thus, we have colored Q_{d+1} with $d + 1$ colors. Since every vertex has degree $d + 1$, we cannot use fewer colors. Therefore, $\chi'(Q_{d+1}) = d + 1$. \square

Problem 4

Prove that $R(3, 3) = 6$, where $R(k, \ell)$ denotes the Ramsey number corresponding to k and ℓ .

Proof. We first claim that there exists a coloring K_5 with neither a red triangle nor a blue triangle, implying that $R(3, 3) > 5$. Consider the following colored K_5 with no red triangles or blue triangles.



Now, we claim that $R(3, 3) \leq 6^5$. Consider a complete graph K_6 where every edge is colored either red or blue. We will show that there exists either a red

⁵We can prove this easier by considering the fact that $R(k, \ell) \leq R(k, \ell - 1) + R(k - 1, \ell)$. Applying the result here yields $R(3, 3) \leq R(3, 2) + R(2, 3)$. Trivially, we know that $R(2, 3) = R(3, 2) = 3$, and thus $R(3, 3) \leq 3 + 3 = 6$. However, from the wording of Problem 5, I assume an explicit argument is desired

triangle or a blue triangle. First, pick any vertex v in the graph. Since the graph is K_6 , vertex v is incident to 5 edges. By the Pigeonhole Principle, since there are 5 edges and only 2 colors, at least $\lceil 5/2 \rceil = 3$ edges incident to v must be of the same color. Assume that there are three edges incident to v that are colored blue (without loss of generality) and let the other endpoints of these edges be u_1, u_2 , and u_3 . Consider the edges connecting u_1, u_2 , and u_3 . There are two possible cases, either at least one edge among $\{u_1, u_2, u_3\}$ is blue or none of the edges among $\{u_1, u_2, u_3\}$ are blue. For the first case, suppose the edge (u_1, u_2) is blue. Since the edges (v, u_1) and (v, u_2) are also blue (by our assumption), and so the vertices $\{v, u_1, u_2\}$ form a blue triangle. For the second case, if none of the edges among $\{u_1, u_2, u_3\}$ are blue, then this implies that all edges connecting these vertices must be red, and so the edges (u_1, u_2) , (u_2, u_3) , and (u_3, u_1) form a red triangle on the vertices $\{u_1, u_2, u_3\}$. In both cases, the graph K_6 contains a monochromatic triangle. Thus, $R(3, 3) \leq 6$. Since $R(3, 3) \leq 6$ and $R(3, 3) > 5$, we have that $R(3, 3) = 6$

□

Problem 5

Show that $R(4, 3) \leq 10$ by proving that every red/blue coloring of the edges of K_{10} contains either a red K_4 or a blue triangle.

Proof. We proceed by contradiction. Assume there exists a 2-coloring (Red and Blue) of the edges of the complete graph K_{10} that contains neither a red K_4 nor a blue triangle (K_3). Let v be an arbitrary vertex in K_{10} . Since the graph has 10 vertices, we have that $\deg(v) = 9$. Let R_v be the set of neighbors connected to v by red edges, and B_v be the set of neighbors connected to v by blue edges. Thus, we have $|R_v| + |B_v| = 9$. Now, we know that $R(3, 3) = 6$ and $R(4, 2) = 4$. For the first case, suppose $|B_v| \geq 4$. The subgraph induced by the vertices in B_v contains at least 4 vertices. Since $R(4, 2) = 4$, this subgraph must contain either a red K_4 or a blue K_2 (a blue edge). If it contains a red K_4 , we have found a red K_4 , else, if it contains a blue edge between two neighbors $u, w \in B_v$, then the vertices $\{v, u, w\}$ form a blue triangle (since edges vu and vw are blue). Since we assumed neither exists, we must have $|B_v| \leq 3$. Now, for the second case, suppose $|R_v| \geq 6$. The subgraph induced by the vertices in R_v contains at least 6 vertices. Since $R(3, 3) = 6$, this induced subgraph must contain either a red

K_3 or a blue K_3 . If it contains a blue K_3 , we have found a blue triangle, else if it contains a red K_3 formed by neighbors $\{x, y, z\}$, then adding v creates the set $\{v, x, y, z\}$ which forms a red K_4 (since edges vx, vy, vz are red). Since we assumed neither exists, we must have $|R_v| \leq 5$. To avoid a contradiction, every vertex v must satisfy both $|B_v| \leq 3$ and $|R_v| \leq 5$. Summing them up yields $|R_v| + |B_v| \leq 5 + 3 = 8$, but we know the degree of v is 9, and $9 \leq 8$, which is a contradiction. Therefore, no such coloring exists, and any 2-coloring of K_{10} must contain either a red K_4 or a blue triangle. Thus, $R(4, 3) \leq 10$. \square

Midterm 1

Problem 1

How many different such trees exist on 100 labeled vertices, labeled v_1, \dots, v_{100} , each of which contains all the edges in the set $\{v_i v_j : i + j = 101\}$?

Problem 2

Let G be a graph on 1863 vertices, each of degree at least 1492. What is the size of the largest matching in G ? (*Reminder: a matching is a set of pairwise disjoint edges.*)

Problem 3

There is a small island with n families living on it. The Ministry of Agriculture has partitioned the land into n equal-sized hunting zones, and independently, also into n equal-sized planting zones. It is the duty of the Ministry of Social Affairs to assign each of the n families one hunting zone and one planting zone. Officials of this ministry are concerned about the independence of the two partitions. They fear that it may not be possible to assign each family a hunting zone and a planting zone in such a way that the hunting zone of each family intersects the planting zone of the same family, which was their original intention. However, the Chief Magician of the island claims that such an assignment is always possible, even without seeing the actual partitions. Prove that he is correct.

Problem 4

Given a directed graph $D = (V, A)$, vertices $s, t \in V$, and conservative cost functions $c_1, \dots, c_k : A \rightarrow \mathbb{R}$. Give an algorithm that decides whether there exists a directed $s-t$ path that is of minimum cost with respect to all c_i .

Problem 5

Let x_1 and x_2 be maximum flows in the network $D = (V, A, s, t, c)$, where $s, t \in V$ and $c : A \rightarrow \mathbb{R}_+$ is a capacity function. Show that there exists

a maximum flow x such that $x(sv) = x_1(sv)$ for every arc leaving s , and $x(wt) = x_2(wt)$ for every arc entering t .

Midterm 2

Problem 1

Let G be a 2-edge connected graph. Show that the number of ears of even length in any ear decomposition of G has the same parity. In other words, although the ear decomposition is not unique, the parity of the number of even ears is an invariant.

Proof.

□

Problem 2

Assume that all sides of a convex polyhedron are either regular pentagons or regular hexagons. What is the number of pentagons among these sides?

Proof.

□

Problem 3

Prove that the inequality $|E(G)| \geq \binom{\chi(G)}{2}$ holds for every finite simple graph G , where $\chi(G)$ denotes the chromatic number of G .

Proof.

□

Problem 4

Let G be a 3-regular graph with edge-chromatic number 3. Assume that (up to permutation of colors) there is only one valid 3-edge-coloring of G . Prove that G contains a Hamiltonian cycle.

Proof.

□

Problem 5

Prove that $R(4, \ell) \leq \binom{\ell+2}{3}$ for all $\ell \geq 2$.

Proof.

□

Table 1: Scores

Set	Scores by Question										Total
Set 1	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	9.0
Max	1	1	1	1	1	1	1	1	1	1	10
Set 2	3.5	1.0	1.0	2.0	-	1.0	0.5	1.0			10.0
Max	4	2	1	2	2	1	1	1			14
Set 3	0.5	2.0	2.0	2.0	-						6.5
Max	2	2	2	2	2						10
Set 4	1.0	0.3	1.5	-	-	2.0	0.1	2.0			6.9
Max	1	1	2	1	1	2	2	2			10
Set 5	1.0	1.0	-	0.2	-	1.0	1.0				4.2
Max	1	1	1	1	1	1	1				7
Set 6	0.5	1.0	-	1.0	-						2.5
Max	1	1	1	1	1						5
Set 7	1.0	4.0	4.0	2.0	1.0						12.0
Max	1	6	4	2	1						14
Set 8	2.0	2.0	2.0	0.5	2.0						8.5
Max	2	2	2	2	2						10
Set 9	1.0	0.5	2.0	2.0	2.0						7.5
Max	1	1	2	2	2						8
Midterm 1											13
Max											50
Midterm 2											12
Max											50
											92.1 (A)