

Due: Tuesday, September 24, 2019 at 10:00 PM

Sundry

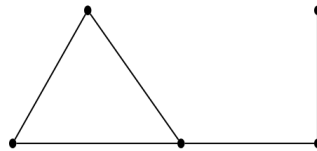
Before you start writing your final homework submission, state briefly how you worked on it. Who else did you work with? List names and email addresses. (In case of homework party, you can just describe the group.)

Note: This homework consists of two parts. The first part (questions 1-6) will be graded and will determine your score for this homework. The second part (questions 7-9) will be graded if you submit them, but will not affect your homework score in any way. You are strongly advised to attempt all the questions in the first part. You should attempt the problems in the second part only if you are interested and have time to spare.

Part 1: Required Problems

1 Degree Sequences

The *degree sequence* of a graph is the sequence of the degrees of the vertices, arranged in descending order, with repetitions as needed. For example, the degree sequence of the following graph is $(3, 2, 2, 2, 1)$.



For each of the parts below, determine if there exists a simple undirected graph G (i.e. a graph without self-loops and multiple-edges) having the given degree sequence. Justify your claim.

(a) $(3, 3, 2, 2)$

(b) $(3, 2, 2, 2, 2, 1, 1)$

- (c) (6, 2, 2, 2)
- (d) (4, 4, 3, 2, 1)

2 Bipartite Graphs

An undirected graph is bipartite if its vertices can be partitioned into two disjoint sets L, R such that each edge connects a vertex in L to a vertex in R (so there does not exist an edge that connects two vertices in L or two vertices in R).

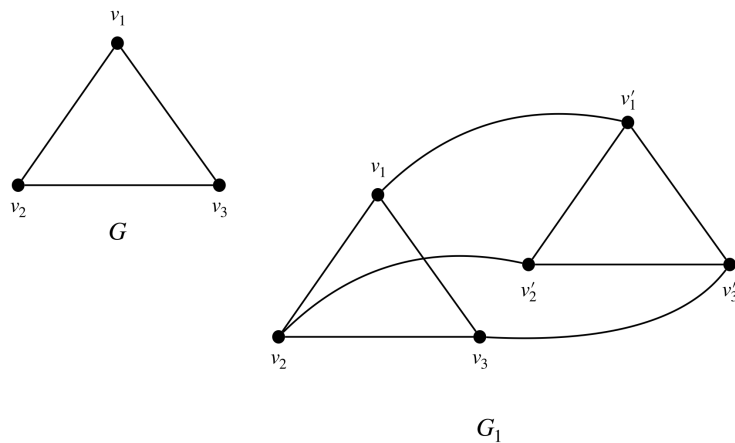
- (a) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Prove that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$.
- (b) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Let s and t denote the average degree of vertices in L and R respectively. Prove that $s/t = |R|/|L|$.
- (c) The *double* of a graph G consists of two copies of G with edges joining the corresponding “mirror” vertices. More precisely, if $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and E the set of edges, then the double of the graph G is given by $G_1 = (V_1, E_1)$, where

$$V_1 = \{v_1, v_2, \dots, v_n, v'_1, v'_2, \dots, v'_n\},$$

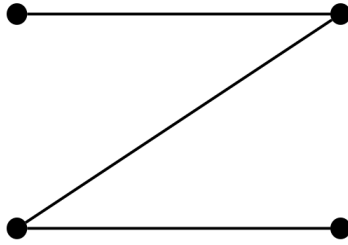
and

$$E_1 = E \cup \{(v'_i, v'_j) \mid (v_i, v_j) \in E\} \cup \{(v_i, v'_i), \forall i\}.$$

Here is an example,



Draw the double of the following graph:



- (d) Now suppose that G_1 is a bipartite graph, G_2 is the double of G_1 , G_3 is the double of G_2 , and so on. (Each G_{i+1} has twice as many vertices as G_i). Show that $\forall n \geq 1, G_n$ is bipartite.

Hint: Use induction on n .

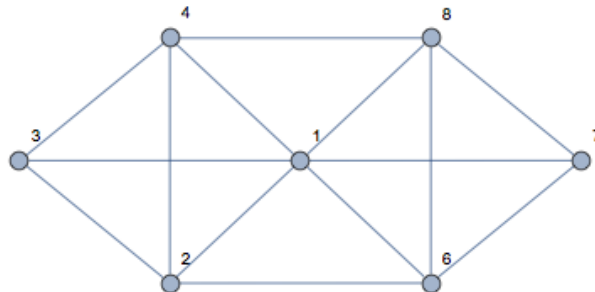
3 Planarity and Graph Complements

Let $G = (V, E)$ be an undirected graph. We define the complement of G as $\bar{G} = (V, \bar{E})$ where $\bar{E} = \{(i, j) | i, j \in V, i \neq j\} - E$; that is, \bar{G} has the same set of vertices as G , but an edge e exists in \bar{G} if and only if it does not exist in G .

- (a) Suppose G has v vertices and e edges. How many edges does \bar{G} have?
- (b) Prove that for any graph with at least 13 vertices, G being planar implies that \bar{G} is non-planar.
- (c) Now consider the converse of the previous part, i.e., for any graph G with at least 13 vertices, if \bar{G} is non-planar, then G is planar. Construct a counterexample to show that the converse does not hold.

Hint: Recall that if a graph contains a copy of K_5 , then it is non-planar. Can this fact be used to construct a counterexample?

4 Eulerian Tour and Eulerian Walk



- (a) Is there an Eulerian tour in the graph above? If no, give justification. If yes, provide an example.

- (b) Is there an Eulerian walk in the graph above? If no, give justification. If yes, provide an example.
- (c) What is the condition that there is an Eulerian walk in an undirected graph? Briefly justify your answer.

5 Trees and Components

- (a) Bob removed a degree 3 node from an n -vertex tree. How many connected components are there in the resulting graph? Please provide an explanation.
- (b) Given an n -vertex tree, Bob added 10 edges to it and then Alice removed 5 edges. If the resulting graph has 3 connected components, how many edges must be removed in order to remove all cycles from the resulting graph? Please provide an explanation.

6 Hamiltonian decompositions of a complete graph

- (a) (Walecki's Theorem) Let K_n be the complete graph on n vertices.
- (i) If n is an odd positive integer, then the edge set of K_n can be partitioned into x edge-disjoint Hamiltonian cycles (a Hamiltonian cycle is a cycle where each vertex appears exactly once).
 - (ii) If n is an even positive integer, then the edge set of K_n can be partitioned into y edge-disjoint Hamiltonian cycles and one perfect matching (a perfect matching is a set of edges such that every vertex is a part of exactly one edge in the set).

Assume the above theorem is true. What are x and y ?

- (b) Give a partition of the edges of K_5 with vertices $\{1, 2, 3, 4, 5\}$ into edge-disjoint Hamiltonian cycles.
- (c) Give a partition of the edges of K_6 with vertices $\{1, 2, 3, 4, 5, 6\}$ into edge-disjoint Hamiltonian cycles and a perfect matching.

Note: This concludes the first part of the homework. The problems below are optional, will not affect your score, and should be attempted only if you have time to spare.

Part 2: Optional Problems

7 Planarity

Consider graphs with the property T : For every three distinct vertices v_1, v_2, v_3 of graph G , there are at least two edges among them. Prove that if G is a graph on ≥ 7 vertices, and G has property

T , then G is nonplanar.

8 Connectivity

Consider the following claims regarding connectivity:

- (a) Prove: If G is a graph with n vertices such that for any two non-adjacent vertices u and v , it holds that $\deg u + \deg v \geq n - 1$, then G is connected.
[Hint: Show something more specific: for any two non-adjacent vertices u and v , there must be a vertex w such that u and v are both adjacent to w .]
- (b) Give an example to show that if the condition $\deg u + \deg v \geq n - 1$ is replaced with $\deg u + \deg v \geq n - 2$, then G is not necessarily connected.
- (c) Prove: For a graph G with n vertices, if the degree of each vertex is at least $n/2$, then G is connected.
- (d) Prove: If there are exactly two vertices with odd degrees in a graph, then they must be in the same connected component (meaning, there is a path connecting these two vertices).
[Hint: Proof by contradiction.]

9 Hypercube Routing

Recall that an n -dimensional hypercube contains 2^n vertices, each labeled with a distinct n bit string, and two vertices are adjacent if and only if their bit strings differ in exactly one position.

- (a) The hypercube is a popular architecture for parallel computation. Let each vertex of the hypercube represent a processor and each edge represent a communication link. Suppose we want to send a packet from vertex x to vertex y . Consider the following “left-to-right bit-fixing” algorithm:

In each step, the current processor compares its address to the destination address of the packet. Let’s say that the two addresses match up to the first k positions (reading the bits from left to right). The processor then forwards the packet and the destination address on to its neighboring processor whose address matches the destination address in at least the first $k + 1$ positions. This process continues until the packet arrives at its destination.

Consider the following example where $n = 4$: Suppose that the source vertex is (1001) and the destination vertex is (0100). Give the sequence of processors that the packet is forwarded to using the left-to-right bit-fixing algorithm.

- (b) The *Hamming distance* $H(x, y)$ between two n -bit strings x and y is the number of bit positions where they differ. Show that for an arbitrary source vertex and arbitrary destination vertex, the number of edges that the packet must traverse under the left-to-right bit-fixing algorithm is the Hamming distance between the n -bit strings labeling source and destination vertices.

- (c) There is another famous graph, called the butterfly network, which is another popular architecture for parallel computation. You will see this network in CS 170 in the context of circuits for implementing the FFT (fast fourier transform). Here is a diagram of the butterfly network for $n = 3$. In general, the butterfly network has $(n + 1) \cdot 2^n$ vertices organized into $n + 1$ columns of 2^n vertices each. The vertices in each column are labeled with the bit strings in $\{0, 1\}^n$, and all vertices in the same row have the same label. The source is on the leftmost column and the destination is on the right.

It turns out the n -butterfly network is equivalent to the n -dimensional hypercube unrolled into n left-to-right bit-fixing steps. On the graph below, label the vertices in the graph explicitly and trace the path from source $x = (110)$ to destination $y = (011)$. Convince yourself that the left to right paths in the butterfly network are indeed equivalent to the hypercube routing obtained by the left-to-right bit-fixing algorithm.

