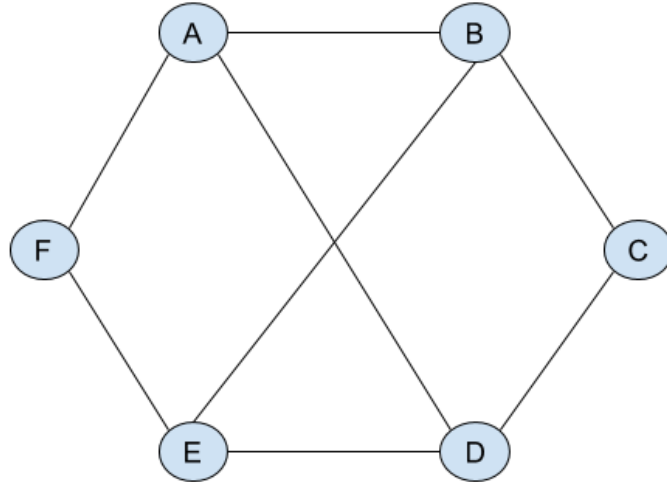


Prepared by: Aishani Sil, Austin Lei, Agnibho Roy, Debayan Bandyopadhyay, Abinav Routhu

1 Graph 101



1. Take a look at the following undirected graph.

(a) How many vertices are in this graph?

Solution: 6

(b) What is the degree of vertex B ?

Solution: 3

(c) What is the total degree of this graph?

Solution: 16

(d) Consider the traversal $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$. How would you categorize it (walk / cycle / path / tour)?

Solution: All of the above.

(e) Give an example of a simple path of length 4.

Solution: $A \rightarrow B \rightarrow C \rightarrow D$ and many others.

(f) Is it possible to construct a traversal that is a tour but not a cycle from this graph (can go through vertices twice, but not edges)? Why or why not?

Solution: No, because there are no vertices with even degree greater than 2 (which is needed to enter and exit a vertex twice and also start and end at the same place).

2. Consider all complete undirected graphs on an even number of vertices. Prove that such graphs can be partitioned into $\frac{n}{2}$ spanning trees that share no edge with another spanning tree.

Solution: We proceed with a proof by induction on the number of vertices.

Base Case: For a 2-vertex complete undirected graph, there is only one spanning tree, which consists of the edge between the two nodes. $1 = \frac{2}{2} = \frac{n}{2}$

Inductive Hypothesis: for some even $k \geq 2$, the k -complete undirected graph can be partitioned into $\frac{k}{2}$ spanning trees that share no edge with another spanning tree.

Inductive Step:

We now consider the complete undirected graph on $k + 2$ vertices. First, let us shrink our $k + 2$ graph by removing 2 vertices (denoted as x and y). Now applying the induction hypothesis, we know there are $\frac{k}{2}$ spanning trees that include all but two vertices. We will now try to build new spanning trees by connecting each of these trees to x and y in the following manner.

First, add back the two new vertices and split the pre-existing k vertices into two halves, A and B . To re-construct the existing $\frac{k}{2}$ spanning trees: for each spanning tree, we will add to it an edge from x to a vertex within A , and an edge from y to a vertex within B . Note that in order to prevent shared edges between spanning trees, we need to pick out a new vertex from A and B each time to connect to x and y . Since the $k + 2$ graph is complete, there will indeed be enough vertices in A and B for us to do this modification for the first $\frac{k}{2}$ spanning trees.

After we have found the first $\frac{k}{2}$ new spanning trees, we need a last one (total of $\frac{k+2}{2} = \frac{k}{2} + 1$). We know that to be a spanning tree, all vertices in A and B must be a part of it, along with x and y of course. Since we have already used some edges from x to set A (same case for y and B), to construct this last tree we want to instead use all the edges from x to vertices in set B , and y to vertices in set A .

The entire $k + 2$ graph is made of vertices either in A , B , or vertices x and y , so the only thing left to connect in this new tree is x and y , which we can do since no previous spanning tree has used this edge.

3. Draw a simple bipartite graph with 6 vertices and 8 edges. What is the most edges you could have in a bipartite graph with 6 vertices? With $2n$?

Solution: The most edges will occur with a complete bipartite graph with 3 vertices on one side and 3 on the other; this has 9 edges. Maximum edges is n^2 . Any bipartite graph on $2n$ vertices makes 2 partitions with size a and b , satisfying $a + b = 2n$. The maximum number of edges given a and b is simply ab (every vertex in A has edge to every vertex in B). This is maximized when $a = b$.

4. Which of these graphs have Eulerian tours?

(a) The complete graph on 5 vertices (K_5).

Solution: Yes, every vertex is of degree 4. A graph in which every vertex has even degree has an Eulerian tour.

(b) The complete graph on 6 vertices (K_6).

Solution: No, every vertex is of degree 5.

(c) The complete graph on 7 vertices (K_7).

Solution: Yes, every vertex is of degree 6.

(d) The 3-dimensional hypercube.

Solution: No, every vertex is of degree 3.

(e) The 4-dimensional hypercube.

Solution: Yes, every vertex is of degree 4.

2 Planarity & Coloring

1. Consider a group of 8 friends sitting at a round table. Any one person is friends with the two individuals next to them and the person sitting directly across from them. Consider the graph where each individual is a vertex and an edge exists between person u and v if and only if u and v are friends. Prove that this graph is non-planar.

Solution: Consider the subgraph of this graph on 6 vertices, partitioned into the left side of the three vertices 0, 3, 5 and the right side of 1, 4, 7. Notice that there is an edge from every node on the left side to the right side and vice versa, creating a $K_{3,3}$ subgraph. Since this graph contains a $K_{3,3}$ subgraph.

2. Show that any tree is 2-colorable.

Solution: We prove the result by induction on the number of vertices.

Base case: $v = 1$

The graph is a single point, which is clearly 2-colorable.

Inductive step: Suppose that the problem statement holds for $v = k$, with $k \geq 1$. We show that the problem statement holds for $v = k + 1$.

Let G be the tree of $k + 1$ vertices, and consider any vertex w in G . Remove w and all the edges connected to it, leaving $\leq k$ connected components. These connected components must also be trees, since if the connected component contained cycle, G would also contain a cycle. By the inductive hypothesis, each of these connected components is 2-colorable. Now, add w back into the graph. w is connected to at most k other vertices. We can rotate the colorings in each of the connected components such that each of the $\leq k$ vertices w is connected to is the same color, allowing us to color w the second color. Thus, G is 2-colorable, as desired.

Thus, the induction is complete.

We can also prove the result by induction on the number of edges:

Base case: $e = 0$

The graph is single point, which is clearly k -colorable.

Inductive step: Suppose that the problem statement holds for $e = k$, with $k \geq 1$. We show that the problem statement

holds for $e = k + 1$.

Let G be the graph of $k + 1$ edges, and consider any edge e in G . Remove e , leaving two connected components. These connected components must also be trees, since if the connected component contained cycle, G would also contain a cycle. By the inductive hypothesis, each of these connected components is 2-colorable. Now, add e back into the graph. e connects two vertices, one from each connected component. We can rotate the colorings in each of the 2 connected components such that the 2 vertices w connects are different colors, giving a 2-coloring of G . Thus, G is 2-colorable, as desired.

Thus, the induction is complete.

3. You are hosting a very exclusive party such that a guest is only allowed to come in if they are friends with you or someone else already at the party. After everyone has showed up, you notice that there are n people (including yourself); each person has at least one friend (of course), but no one is friends with everyone else. It is still quite a sad party, because among all the possible pairs of people, there are only a total of $n - 1$ friendships. You want to play a game with two teams, and in order to kindle new friendships, you want to group the people (including yourself) such that within each team, no one is friends with each other. Is this possible? (*Hint: How might the previous question be useful?*)

Solution: Represent people as vertices and friendships between them as edges on a graph G . Based on the circumstances surrounding the party's guests, we know that G is connected (every new friend/vertex needs to be connected to a vertex within the existing graph) and there are a total of $n - 1$ edges (friendships). G is, in fact, a tree!

Now, we want to separate the partygoers into 2 groups such that no two people from the same group are friends. This can be modelled in G by partitioning the vertices into 2 groups such that the only edges are between the groups, not within. Thus, we have reduced the problem to showing a tree G is bipartite. But, we just showed that trees can always be 2-colored! We can just divide the vertices in 2 sets based on what color they are. Thus, all trees are also bipartite.

4. Two knights are placed on diagonally opposite corners of a chessboard, one white and the other black. The knights take turn moving as in standard chess, with white moving first.
- (a) Let every square on a chessboard represent a vertex of a graph, with edges between squares that are a knight's move away. Describe a 2-coloring of this graph.

Solution: Color white squares white, and black squares black, in a checkerboard pattern. Since knights always go from a white square to a black one and vice versa, this forms a 2-coloring.

- (b) Show that the black knight can never be captured, even if it cooperates with the white knight.

Solution: At the end of the black knight's move, the two knights are on the same color. Thus, when the white knight moves, it lands on the opposite color of the black knight, and so can never capture the black knight.

3 Hypercubes

1. Austin has n lightbulbs in a row, all turned off. Every second Austin performs a **move**, where Austin either turns a lightbulb on or turns a lightbulb off. Show that for any n , there is a sequence of moves that Austin can make such that each possible configuration of lightbulbs being on or off has occurred exactly once, and such that the last move causes all the lightbulbs to be off.

Solution: We can imagine the situation as an a cycle on an n -dimensional hypercube, where each vertex is labeled as a length n bit string. The cycle starts at $0 \dots 0$ and travels to every vertex. (This is called a **Hamiltonian cycle**.) We show

that such a cycle exists for all n by induction.

Base case: $n = 1$

The graph consists of the vertices 0 and 1 connected by an edge, which has the Hamiltonian cycle starting at 0, then going to 1 then back to 0.

Inductive step: Suppose that the problem statement holds for $n = k$, with $k \geq 1$. We show that the problem statement holds for $n = k + 1$.

Let H be our $k + 1$ -dimensional hypercube. Define H_0 to be the k -dimensional hypercube constructed by all the vertices in H with first bit 0, and H_1 to be the k -dimensional hypercube formed by all the vertices in H with first bit 1. By the inductive hypothesis, there is a Hamiltonian cycle of H_0 starting at $00 \dots 0$. Let $00 \dots 010 \dots 0$ be the last vertex in the Hamiltonian cycle before $00 \dots 0$. Similarly, by the inductive hypothesis, there is a Hamiltonian cycle of H_1 starting at $10 \dots 0$, where $10 \dots 010 \dots 0$ is the last vertex in the Hamiltonian cycle before $10 \dots 0$.

Now we construct a new cycle of H . Our new Hamiltonian cycle takes the path from $00 \dots 0$ to $00 \dots 010 \dots 0$ as defined by the Hamiltonian cycle in H_0 , traverses the edge from $00 \dots 010 \dots 0$ to $10 \dots 010 \dots 0$ and , takes the path from $10 \dots 010 \dots 0$ to $10 \dots 0$ as defined by traversing the Hamiltonian cycle in H_1 in reverse, then traversing the edge from $10 \dots 0$ to $00 \dots 0$. This new cycle traverses all the vertices (by the inductive hypothesis), so it is a Hamiltonian cycle of H , as desired.

Thus, the induction is complete.

2. You wish to color the *edges* of a n -dimension hypercube, such that edges that share a vertex are different colors. (Note: This differs from the normal definition of graph coloring, where you color vertices so that vertices that share an edge are different colors). Prove that n colors are both sufficient and necessary; that is, you can do this with n colors, but not with $n - 1$ colors.

Solution: Consider the bitwise representation of each vertex. Every vertex is connected to n others, so it is impossible to use less than n colors; otherwise each vertex will have multiple edges with the same color.

Color all edges which go between vertices $0xxxx\dots$ and $1xxxx\dots$ one color, with $xxxx\dots$ ranging over all sequences of $n - 1$ bits. None of these edges share a vertex, so our requirement is preserved. Furthermore, we can do the same with each "position" in the bitwise representation; thus, edges connecting vertices $x0xxx\dots$ to $x1xxx\dots$ will share a second color, edges connecting vertices $xx0xx\dots$ to $xx1xx\dots$ will share a third color, and so on. This overall uses n colors, and encompasses all edges in the hypercube, while ensuring that edges that share a vertex are different colors. Thus, this forms an n coloring of the edges.