Recitation Module 12

Penn Engineering

Lecture Review

- Isomorphic graphs

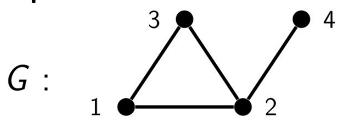
Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic**, notation $G_1 \simeq G_2$, when there is a **bijection** $\beta : V_1 \to V_2$ such that for any $u_1, v_1 \in V_1$ we have $u_1\text{–}v_1 \in E_1$ **iff** $\beta(u_1)\text{–}\beta(v_1) \in E_2$.

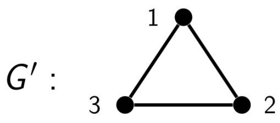
- Subgraphs and induced subgraphs

A graph $G_1 = (V_1, E_1)$ is a **subgraph** of the graph $G_2 = (V_2, E_2)$ when $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$. (Beware: not all pairs of such subsets form graphs!)

If $G = (V, E)$ is a graph and $V' \subseteq V$ is a set consisting of some of G's nodes, the subgraph of G **induced** by V' is the graph $G' = (V', E')$ where E' consists of all the edges of G whose endpoints are both in V'.

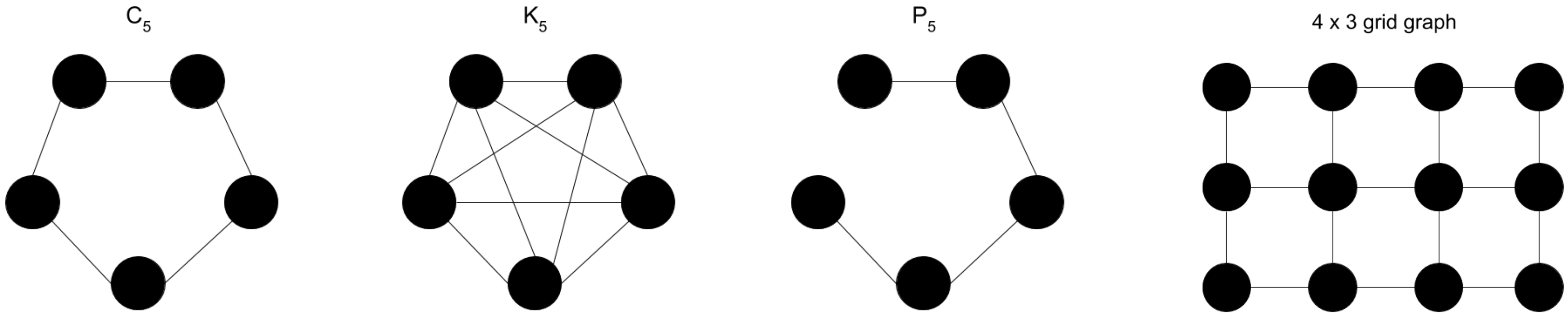
Example:





Lecture Review

- Cycle, complete, path, and grid graphs



- Closed walks and cycles

A **closed walk** is a walk in which the first and the last vertex are the same.

A **cycle** is a closed walk **of length at least 3** in which all nodes are pairwise distinct, except for the last and the first.
The **length** of the cycle is the length of the closed walk.

Basic terms/definitions to know!!

- Acyclic graphs, trees, and forests

A graph in which there are no cycles is called **acyclic**. The cc's of an acyclic graph are also acyclic.

A graph that is both connected and acyclic is called a **tree**.

Consequently, an acyclic graph is also called a **forest** since all its cc's are trees!

- Cut edges

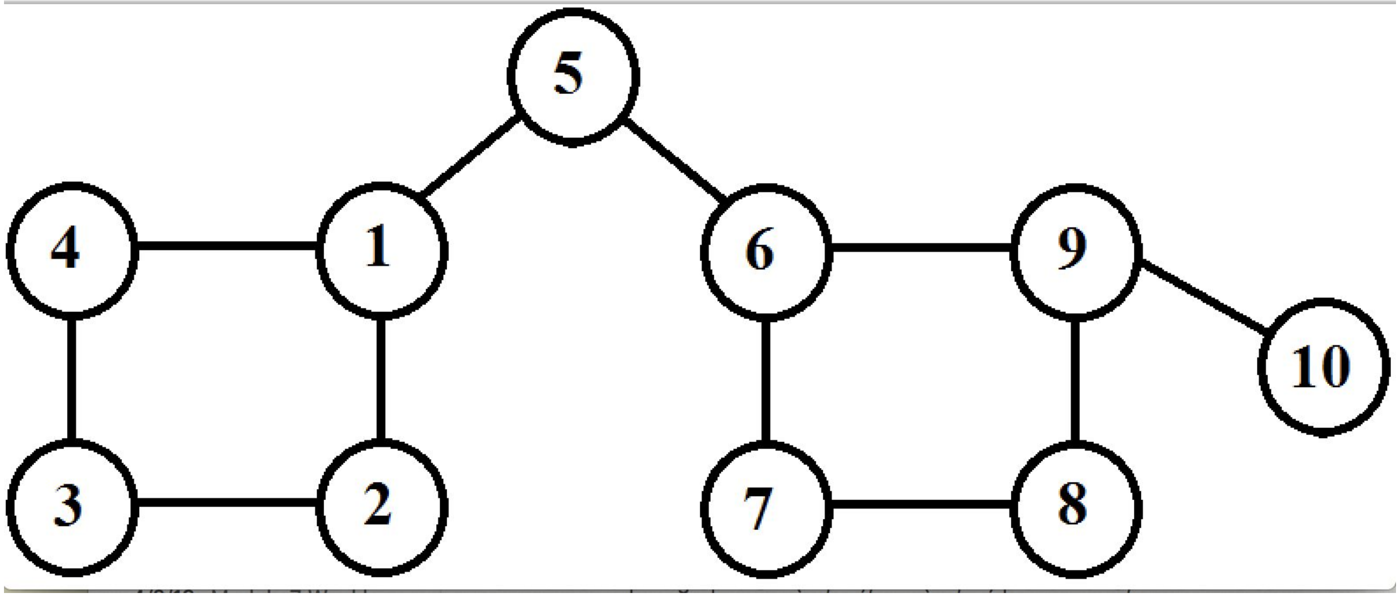
Let $G = (V, E)$ be a graph. An edge of G is a **cut edge** if by removing it we obtain a graph with strictly **more** connected components (cc's) than G.

Basic terms/definitions to know!!

- Properties of trees
  - Every tree is <u>minimally connected</u>:
    - i.e. removing any edge in a tree disconnects it
  - Every tree is <u>maximally acyclic</u>:
    - Adding an edge between **any** two non adjacent vertices in a tree creates a cycle
    - Adding an edge to an acyclic graph creates at most one cycle
  - $|E| = |V| - 1$ (In a forest, $|E| = |V| - |CC|$)
  - Any two distinct vertices of a tree are connected by a **unique** path
- Every tree has at least one leaf which is vertex of degree 1.

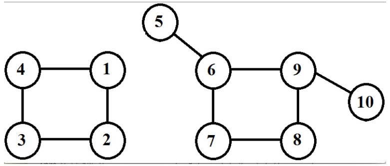
Question 1

Identify the cut edges in the following graph:

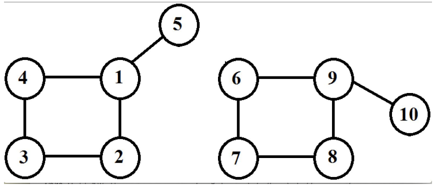


Answer to Question 1

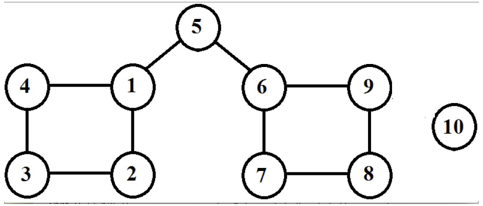
A. Edge **5-1**



B. Edge **5-6**



C. Edge **9-10**



Question 2

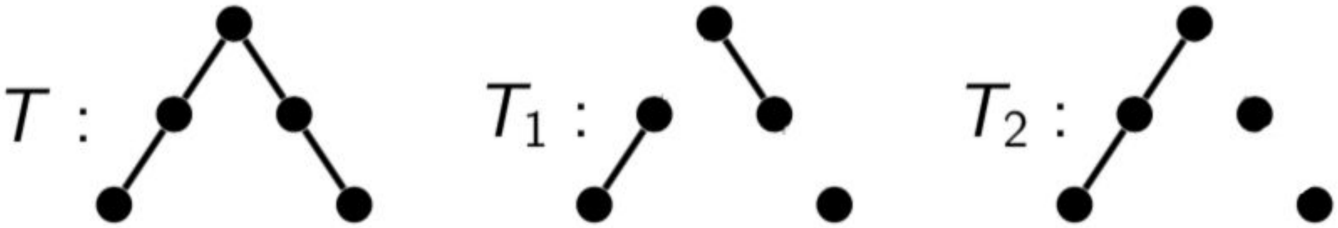
Consider a tree with 5 nodes. Remove 2 edges.

a. How many CCs will the resulting graph have?

b. Why must at least one of the resulting CCs have at least 2 nodes?

Answer to Question 2a

Recall that removing any edge from a tree disconnects it. Consider the following example:



We can see that the resulting graph has 3 connected components.

This is because when we remove an edge from a tree it becomes disconnected (2 CCs)
The resulting CCs are also trees. So when we remove an edge from one of the CCs it becomes 2 CCs. Thus in total we have 3 CCs.

Answer to Question 2b

Since a tree is by definition minimally connected, when we remove one edge it splits into two connected components. And since removing an edge cannot create a cycle, both connected components are trees. That means they are minimally connected, and thus, when we remove one edge from either of those connected components, that connected component will split into two, giving us a total of 3 connected components after we remove two edges.

In the first split, one component will have 3 or more nodes (by PHP). If an edge is removed from other component, this one will have at least 2 nodes. If the second split is also on this component, it’s guaranteed to result in a new component with 2 or more nodes due to PHP.

Question 3

How many paths of length at least 1 are in C_{100} and K_{100}?

Answer to Question 3

In C_{100}, there are two paths between every pair of nodes (one clockwise, and one counter clockwise), so there are 2 * (100 choose 2) = 2 * 4950 = 9900 paths of length at least 1.

In K_{100}, we can get from any vertex to any other vertex directly. After picking distinct start and end nodes, every permutation of any quantity of the remaining vertices gives a path.

Thus the number of paths of length at least 1 with given start and end nodes is equal to

$$\sum_{k=0}^{98} \binom{98}{k} k! = \sum_{k=0}^{98} \frac{98!}{(98-k)!} = 98! \sum_{k=0}^{98} \frac{1}{(98-k)!} = 98! \sum_{k=98}^{0} \frac{1}{(k)!} = 98! \sum_{k=0}^{98} \frac{1}{(k)!}$$

This value approximates to 98! * e because the sum of 1/k! from k = 1 to infinity is equal to e. Finally, to find the total number of paths of length at least 1 in K_{100}, multiply by (100 choose 2).

Do not worry, you will not be asked to compute such approximations. You can just leave the summations.

Before you go...

Do you have any general questions about this week's assignment? We can't go into minute details about any specific questions, but this is the time to ask any general clarifying/confusing questions!

As always, if you can't talk to us during recitation or office hours, post any questions/anything course related on the forums or email mcitonline@seas.upenn.edu. Ask questions that might be beneficial to other students on the forums, while emailing about more personal questions (regrade requests, etc).