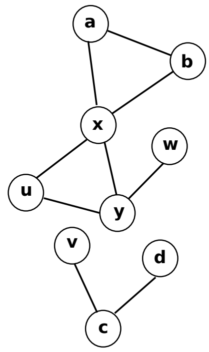
# CS311H: Discrete Mathematics

## Graph Theory II

Instructor: Işıl Dillig



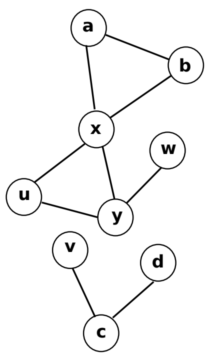
## Connectivity in Graphs




- Typical question: Is it possible to get from some node $u$ to another node $v$?
- Example: Train network – if there is path from $u$ to $v$, possible to take train from $u$ to $v$ and vice versa.
- If it's possible to get from $u$ to $v$, we say $u$ and $v$ are connected and there is a path between $u$ and $v$



## Paths




- A path between $u$ and $v$ is a sequence of edges that starts at vertex $u$, moves along adjacent edges, and ends in $v$.
- Example: $u, x, y, w$ is a path, but $u, y, v$ and $u, a, x$ are not
- Length of a path is the number of edges traversed, e.g., length of $u, x, y, w$ is 3
- A simple path is a path that does not repeat any edges
- $u, x, y, w$ is a simple path but $u, x, u$ is not

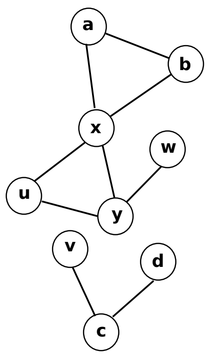


## Example

- Consider a graph with vertices $\{x, y, z, w\}$ and edges $(x, y), (x, w), (x, z), (y, z)$
- What are all the simple paths from $z$ to $w$?
- What are all the simple paths from $x$ to $y$?
- How many paths (can be non-simple) are there from $x$ to $y$?



## Connectedness



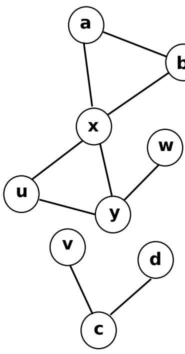

- A graph is connected if there is a path between every pair of vertices in the graph
- Example: This graph not connected; e.g., no path from $x$ to $d$
- A connected component of a graph $G$ is a maximal connected subgraph of $G$



## Example

- Prove: Suppose graph $G$ has exactly two vertices of odd degree, say $u$ and $v$. Then $G$ contains a path from $u$ to $v$.
-
-
-
-

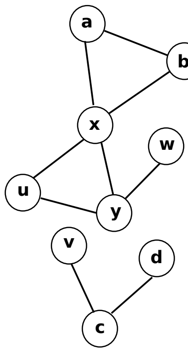
## Circuits




- A circuit is a path that begins and ends in the same vertex.
- $u, x, y, x, u$ and $u, x, y, u$ are both circuits
- A simple circuit does not contain the same edge more than once
- $u, x, y, u$ is a simple circuit, but $u, x, y, x, u$ is not
- Length of a circuit is the number of edges it contains, e.g., length of $u, x, y, u$ is 3
- In this class, we only consider circuits of length $3$ or more



## Cycles

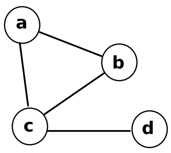



- A cycle is a simple circuit with no repeated vertices other than the first and last ones.
- For instance, $u, x, a, b, x, y, u$ is a circuit but not a cycle
- However, $u, x, y, u$ is a cycle



## Example

- Prove: If a graph has an odd length circuit, then it also has an odd length cycle.
- Huh? Recall that not every circuit is a a cycle.
- According to this theorem, if we can find an odd length circuit, we can also find odd length cycle.
- Example: $d, c, a, b, c, d$ is an odd length circuit, but graph also contains odd length cycle

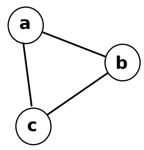





## Proof

Prove: If a graph has an odd length circuit, then it also has an odd length cycle.

- Proof by strong induction on the length of the circuit.
-




-



## Proof, cont.

Prove: If a graph has an odd length circuit, then it also has an odd length cycle.

-
-
-
-



## Proof, cont.

Prove: If a graph has an odd length circuit, then it also has an odd length cycle.

-
-
-
-

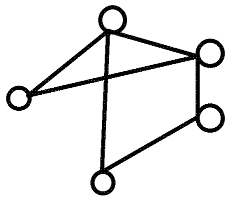
## Colorability and Cycles

Prove: If a graph is 2-colorable, then all cycles are of even length.

- 
- 
- 
- 
- 



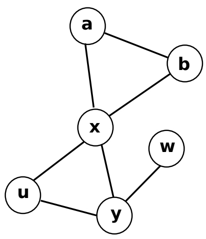
## Example



- Is this graph 2-colorable?



## Distance Between Vertices




- The distance between two vertices $u$ and $v$ is the length of the shortest path between $u$ and $v$
- What is the distance between $u$ and $b$?
- What is the distance between $u$ and $x$?
- What is the distance between $x$ and $w$?



## More Colorability and Cycles

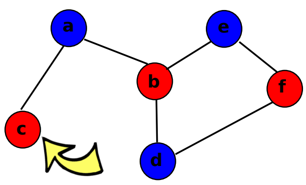
Prove: If graph has no odd length cycles, then graph is 2-colorable.

- To prove this, we first consider an algorithm for coloring the graph with two colors.
- Then, we will show that this algorithm works if graph does not have odd length cycles.



## The Algorithm

- Pick any vertex $v$ in the graph.
- If a vertex $u$ has odd distance from $v$, color it blue
- Otherwise, color it red



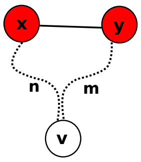



## Proof

- We will now prove: "If the graph does not have odd length cycles, the algorithm is correct."
- Correctness of the algorithm implies graph is 2-colorable.
- Proof by contradiction.
- Suppose graph does not have odd length cycles, but the algorithm produces an invalid coloring.
- Means there exist two vertices $x$ and $y$ that are assigned the same color.

## Proof, cont.

- Case 1: They are both assigned red




- We know $n, m$ are both even
- This means we now have an odd-length circuit involving $n, m$
- By theorem from earlier, this implies that graph has odd length cycle, i.e., contradiction
- Case 2 is exactly the same.



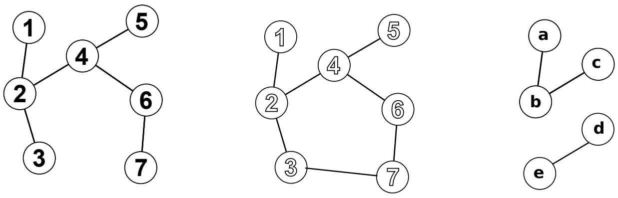
## Putting It All Together

- Theorem: A graph is 2-colorable if and only if it does not have odd-length cycles
- Corollary: A graph is bipartite if and only if it does not have odd-length cycles
- Example: Consider a graph $G$ with vertices $a, b, c, d, e, f$
  - Is G partitle if its edges are $(a, f), (e, f), (e, d), (c, d), (a, c)$?



## Trees

- A tree is a connected undirected graph with no cycles.
- Examples and non-examples:




- An undirected graph with no cycles is a forest.



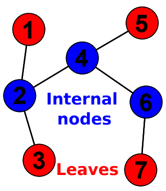
## Fact About Trees

Theorem: An undirected graph $G$ is a tree if and only if there is a unique simple path between any two of its vertices.



## Leaves of a Tree

- Given a tree, a vertex of degree 1 is called a leaf.




- **Important fact: Every tree with more than 2 nodes has at least two leaves.**



## Why is this true?

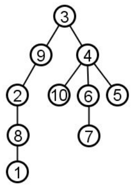
## Number of Edges in a Tree

Theorem: A tree with $n$ vertices has $n-1$ edges.

- Proof is by induction on $n$
- Base case: $n = 1$ ✓
- Induction: Assume property for tree with $n$ vertices, and show tree $T$ with $n+1$ verices has $n$ edges
- Construct $T'$ by removing a leaf from $T$; $T'$ is a tree with $n$ vertices (tree because connected and no cycles)
- By IH, $T'$ has $n-1$ edges
- Add leaf back: $n+1$ vertices and $n$ edges



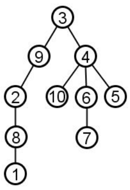
## Rooted Trees




- A rooted tree has a designated root vertex and every edge is directed away from the root.
- Vertex $v$ is a parent of vertex $u$ if there is an edge from $v$ to $u$; and $u$ is called a child of $v$
- Vertices with the same parent are called siblings
- Vertex $v$ is an ancestor of $u$ if $v$ is $u$'s parent or an ancestor of $u$'s parent.
- Vertex $v$ is a descendant of $u$ if $u$ is $v$'s ancestor



## Questions about Rooted Trees

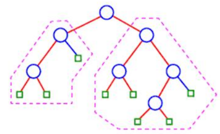



- Suppose that vertices $u$ and $v$ are siblings in a rooted tree.
- Which statements about $u$ and $v$ are true?
  1. They must have the same ancestors
  2. They can have a common descendant
  3. If $u$ is a leaf, then $v$ must also be a leaf



## Subtrees

- Given a rooted tree and a node $v$, the subtree rooted at $v$ includes $v$ and its descendants.



- Level of vertex $v$ is the length of the path from the root to $v$.
- The height of a tree is the maximum level of its vertices.



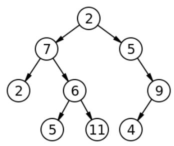
## True-False Questions

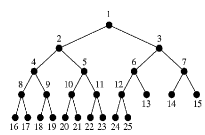
1. Two siblings $u$ and $v$ must be at the same level.
2. A leaf vertex does hot have a subtree.
3. The subtrees rooted at $u$ and $v$ can have the same height only if $u$ and $v$ are siblings.
4. The level of the root vertex is $1$.



## $m$-ary Trees

- A rooted tree is called an $m$-ary tree if every vertex has no more than $m$ children.
- An $m$-ary tree where $m = 2$ is called a binary tree.
- A full $m$-ary tree is a tree where every internal node has exactly $m$ children.
- Which are full binary trees?



## Useful Theorem

Theorem: An $m$-ary tree of height $h \geq 1$ contains at most $m^h$ leaves.

- Proof is by induction on height $h$.
-
-
-
-



## Corollary

Corollary: If $m$-ary tree has height $h$ and $n$ leaves, then $h \geq \lceil log_m n \rceil$

-
-
-



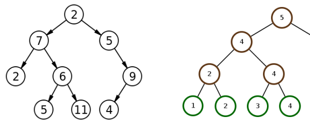
## Questions

- What is maximum number of leaves in binary tree of height 5?
- If binary tree has 100 leaves, what is a lower bound on its height?
- If binary tree has 2 leaves, what is an upper bound on its height?



## Balanced Trees

- An $m$-ary tree is balanced if all leaves are at levels $h$ or $h-1$




- "Every full tree must be balanced." – true or false?
- "Every balanced tree must be full." – true or false?