Algorithms and Data Structures

Exercise Session 9

https://n.ethz.ch/~ahmala/and

Quiz

- incident
- adjacent
- DAG
- Indegree
- Outdegree
- reachable

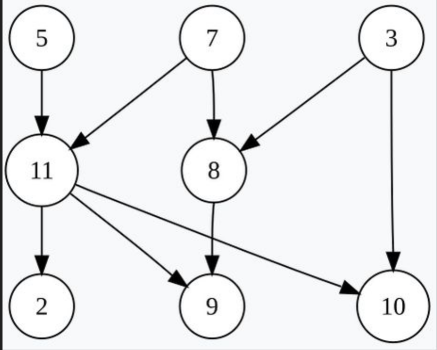
Check with DFS if G is connected

Check with DFS if G is acyclic

Check with DFS if G is a tree

You have to complete n courses. There are m requirements of the form "course a has to be completed before course b". Your task is to find an order in which you can complete the courses.

Topological Sorting

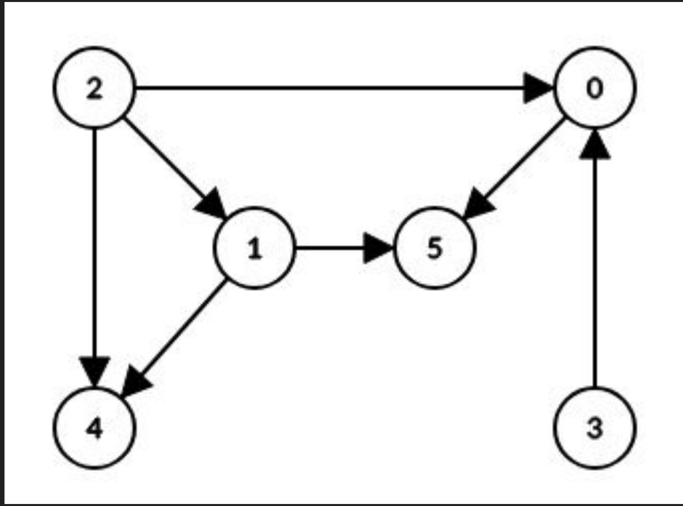


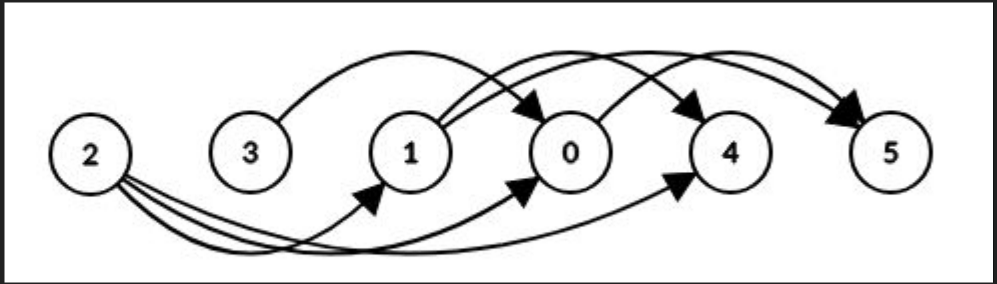
This graph has many valid topological sorts, including:

- 5, 7, 3, 11, 8, 2, 9, 10 (visual left-to-right, top-to-bottom)
- 3, 5, 7, 8, 11, 2, 9, 10 (smallest-numbered available vertex first)
- 3, 5, 7, 8, 11, 2, 10, 9 (lexicographic by incoming neighbors)
- 5, 7, 3, 8, 11, 2, 10, 9 (fewest edges first)
- 7, 5, 11, 3, 10, 8, 9, 2 (largest-numbered available vertex first)
- 5, 7, 11, 2, 3, 8, 9, 10 (attempting top-to-bottom, left-to-right)
- 3, 7, 8, 5, 11, 10, 2, 9 (arbitrary)

Task-1

Find the length of longest path in a directed acyclic graph.





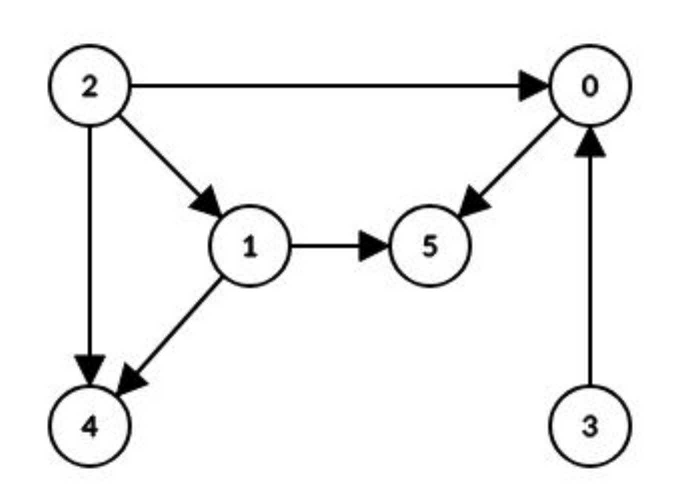
Let $dp[v]$ denote the length of the longest path ending at the node v. Clearly

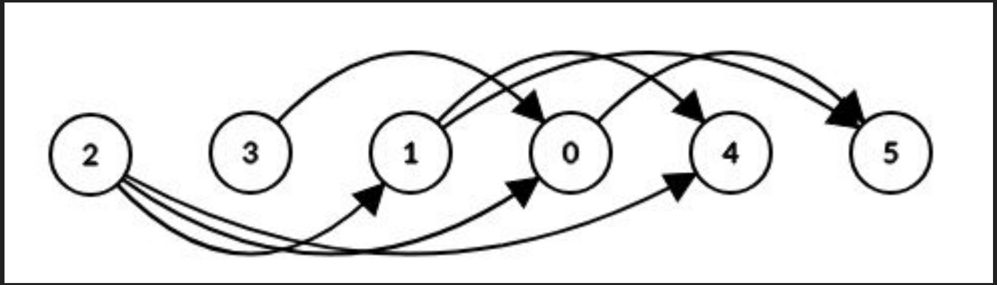
$$dp[v] = \max_{\text{edge } u \to v \text{ exists}} dp[u] + 1,$$

or 1 if v is node 1. If we process the states in topological order, it is guaranteed that $dp[u]$ will already have been computed before computing $dp[v]$.

Task-2

A game has n levels, connected by m teleporters, and your task is to get from level 1 to level n. The game has been designed so that there are no directed cycles in the underlying graph. In how many ways can you complete the game?





A game has n levels, connected by m teleporters, and your task is to get from level 1 to level n. The game has been designed so that there are no directed cycles in the underlying graph. In how many ways can you complete the game?

Let $dp[v]$ denote the number of paths reaching v. We can see,

$$dp[v] = \sum_{\text{edge } u \to v \text{ exists}} dp[u],$$

with an exception of $dp[1]$, or the starting node, having a value of 1. We process the nodes topologically so $dp[u]$ will already have been computed before $dp[v]$.

G = (V,E) is a graph with |V| >= 1 Nodes. Following Expressions are equivalent!

a) G is a tree
b) G is connected with no cycles
c) G is connected with |E| = |V| - 1
d) G has no cycles and |E| = |V| - 1
e) For all x,y in V: G has exactly one x-y-Path

Exercise 8.1 *Introduction to graphs* **(1 point)**.

A group of $n \geq 3$ people wants to play the following *telephone game*: A random player in the group is given a message. The goal is to communicate this message to each member of the group. Each player is allowed to make one phone call and receive one phone call. Furthermore, a player can only make calls to other players who are in their *contact list* (you may assume that if player a is in the contact list of player b, then player b is also in the contact list of player a).

In this exercise, we care about the following question: *under what circumstances is it possible for the group to win the game, regardless of the starting player?* You may assume the group communicated a strategy beforehand, and each player is aware of the contents of each other player's contact list.

(a) Model the telephone game using a graph. Indicate carefully what the vertices and edges of this graph are. Then, give a necessary and sufficient condition for the game to be winnable (regardless of the starting player) using terminology from the lecture. Briefly argue the correctness of your condition.

(b) Give an example of a situation where the game is winnable for some, but not all starting players. Describe your example by drawing the graph that models it according to part (a).

(c) Someone claims the game is always winnable if the following conditions hold:

- Each player has at least two other players in their contact list;
- For any two players a and b, it is possible for the message to reach player b if player a was the starting player.

Translate these conditions to your graph model of part (a) using terminology from the lecture. Then show the claim is false when $n = 5$.

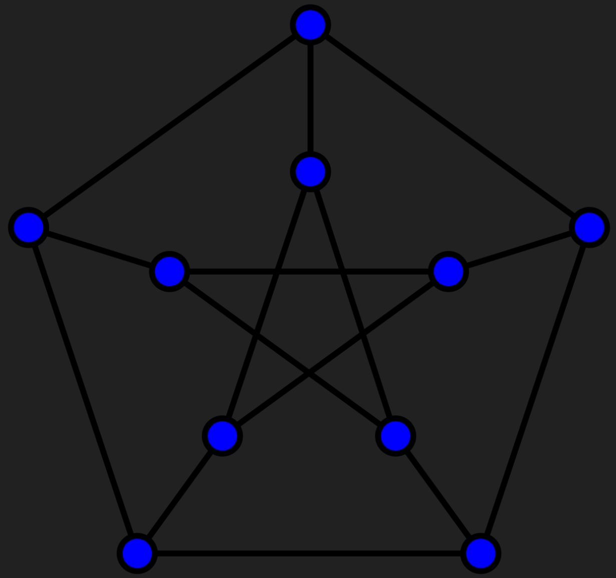
(d)* In a variant of the game for advanced players, the last person to learn the message has to call back the starting player to let them know everything went according to plan. Model this *advanced telephone game* using a graph as in part (a). Then, show that even if the (normal) telephone game is winnable regardless of the starting player, this does not mean the advanced telephone game is also winnable.

Hint: Look up the *Petersen graph*.

Petersen Graph

Hamiltonian path starting from every vertex

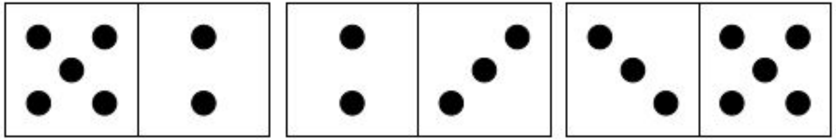
No hamiltonian cycle



Exercise 8.2 *Domino.*

(a) A domino set consists of all possible $\binom{6}{2} + 6 = 21$ different tiles of the form $[x|y]$, where x and y are numbers from $\{1, 2, 3, 4, 5, 6\}$. The tiles are symmetric, so $[x|y]$ and $[y|x]$ is the same tile and appears only once.

Show that it is impossible to form a line of all 21 tiles such that the adjacent numbers of any consecutive tiles coincide like in the example below.



(b) What happens if we replace 6 by an arbitrary $n \geq 2$? For which n is it possible to line up all $\binom{n}{2} + n$ different tiles along a line?

Exercise 8.3 *Star search, reloaded.*

A *star* in an undirected graph $G = (V, E)$ is a vertex that is adjacent to all other vertices. More formally, $v \in V$ is a star if and only if $\{\{v, w\} \mid w \in V \setminus \{v\}\} \subseteq E$.

In this exercise, we want to find a star in a graph G by walking through it. Initially, we are located at some vertex $v_0 \in V$. Each vertex has an associated flag (a Boolean) that is initially set to `False`. We have access to the following constant-time operations:

- `countNeighbors()` returns the number of neighbors of the current vertex
- `moveTo(`i`)` moves us to the ith neighbor of the current vertex, where $i \in \{1..$`countNeighbors()`$\}$
- `setFlag()` sets the flag of the current vertex to `True`
- `isSet()` returns the value of the flag of the current vertex
- `undo()` undoes the latest action performed (the movement or the setting of last flag)

Assume that G has exactly one star and $|V| = n$. Give the pseudocode of an algorithm that finds the star, i.e., your algorithm should always terminate in a configuration where the current vertex is a star in G. Your algorithm must have complexity $O(|V| + |E|)$, and must not introduce any additional datastructures (no sets, no lists etc.). Show that your algorithm is correct and prove its complexity. The behavior of your algorithm on graphs that do not contain a star or contain several stars can be disregarded.

Exercise 8.4 *Introduction to Trees.*

In this exercise the goal is to prove a few basic properties of trees (for the definition of a tree, see Definition 1).

(a) A **leaf** is a vertex with degree 1. Prove that in every tree G with at least two vertices there exists a leaf.

Hint: *Consider the longest path in G. Prove that its endpoint is a leaf.*

(b) Prove that every tree with n vertices has exactly $n-1$ edges.

Hint: *Prove the statement by using induction on* n. *In the induction step, use part (a) to find a leaf. Disconnect the leaf from the tree and argue that the remaining subgraph is also a tree. Apply the induction hypothesis and conclude.*

(c) Prove that a graph with n vertices is a tree if and only if it has $n-1$ edges and is connected.

Hint: *One direction is immediate by part (b). For the other direction (every connected graph with* $n-1$ *edges is a tree), use induction on* n*. First, prove there always exists a leaf by considering the average degree. Then, disconnect the leaf from the graph and argue that the remaining graph is still connected and has exactly one edge less. Apply the induction hypothesis and conclude.*

8.5 (Bonus Task)

(a) Let $v \neq w \in V$ and suppose that G is a tree. Prove that if P_1 and P_2 are paths that both start at v and end at w, then $P_1 = P_2$.

(b) If every vertex of G has at least $\lceil n/2 \rceil$ neighbors, then G is connected

(c) If G contains a Hamiltonian cycle C, then any other Hamiltonian cycle of G must contain an edge from C.

(d) For every graph G with $n \geq 2$, there must be at least two vertices with the same degree.

(e) Suppose in a graph G, for every path of length at least 2, the sum of the degrees of the vertices in the path is even. Then G has an Eulerian walk.

(f) Let G be a connected graph. Suppose that deleting any edge of G does not disconnect the graph. Then deleting any vertex of G does not disconnect the graph. (When deleting a vertex, we also remove all edges incident to the vertex.)

Peer Grading

Exercise 8.1