Plan

- Administrative Information
- Discussion of Assignment 7
- Preview of Assignment 9
- Recap DFS
- Graph Quizzes
- Peer Grading (Ex. 8.1)

Administrative Information

- graph-cheatsheet updated (DFS, directed graphs)
  ↳ can be found in the Polybox

- going forward: no exceptions for missing the peer-grading deadlines
  ↳ Hand-in until Tuesday 12:00
  ↳ make sure your group actually sent it (talk to your partner!)

- if you type your solutions on the Computer:
  ↳ use the same mathematical symbol as in the exercise, not the "most similar letter"
    ↳ e.g. $\varepsilon \neq E$, $\mathbb{N} \neq N$
  ↳ use underscore for subscripts, caret symbol for superscripts, to avoid confusion
    ↳ e.g. 2^k = 2^k, w_i = w_i, $\log_2(a^2)$ = log_2(a^2)
  (For those interested: read into LaTeX, will also help in your later studies ☺)

Discussion of Assignment 7

- 7.1: -Justification for the entry computation was often missing
  -Bounds for the entry computation not (sufficiently stated)

- 7.4: -Missing the runtime justification
  -Remark: When speaking about 3d DP-Tables we can't talk about rows and columns since these terms are only defined for matrices/vectors (1d/2d DP-Tables)

- 7.5: -Not much to say, was solved very well!

Preview of Assignment 9

- only graph exercises
- 3/4 are bonus
  ↳ try to do the non-bonus (doesn't take long)

Depth-First-Search (DFS)

- One of the most important algorithms (together with BFS)
  - ↳ very wide range of use (not just in AnD ◡̈)
  - ↳ learn to implement this by heart, you will be using it a lot
- Runtime $O(|V|+|E|)$ (with adjacency lists)

Algorithm 3 Visit(u)

pre[u] $\leftarrow T$; $T \leftarrow T+1$
markiere u
for Nachvolger v von u, unmarkiert **do**
    Visit(v)
post[u] $\leftarrow T$; $T \leftarrow T+1$

Algorithm 4 DFS(G)

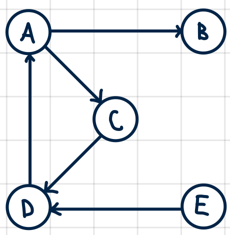
$T \leftarrow 1$
alle Knoten unmarkiert
for $u_0 \in V$, unmarkiert **do**
    Visit(u_0)

- To check for connected components of $u \in V$ (undirected graphs)
  - ↳ run Visit(u)
  - ↳ $v \in ZHK(u) \Rightarrow v$ flagged (=markiert) after Visit(u)

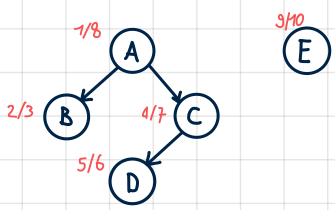
- Using the pre/post-ordering (directed graphs)
  - if $G=(V,E)$ is acyclic $\Rightarrow$ reversed post-order is a topological ordering

  - classification of edges ($e=(u,v)\in E$)
    - edge is in tree $\Rightarrow$ tree edge
    - edge is not in tree
      - ↳ $pre[v] < pre[u] < post[u] < post[v] \Rightarrow$ back edge
      - ↳ $pre[u] < pre[v] < post[v] < post[u] \Rightarrow$ forward edge
      - ↳ $pre[v] < post[v] < pre[u] < post[u] \Rightarrow$ cross edge
      - ↳ other options aren't possible
    - $\exists$ back edge $\Rightarrow \exists$ directed cycle

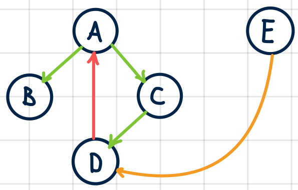
Example:



Recursion/DFS-Tree:



Edge Classification:



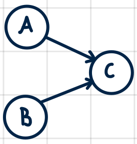
Graph Quizzes (HS20/HS21/HS22)

- Only questions you can answer with your current knowledge
- In exams −1 for wrong answer, 0 for unanswered, 1 for correct answer
- Usually no justification needed

HS20

	Claim	true	false
1	The topological ordering of a directed acyclic graph is unique.	☐	☐
2	For all $n \in \mathbb{N}$, there exists a directed acyclic graph on n vertices with $\binom{n}{2}$ edges.	☐	☐
3	Let $v \in V$ be a vertex of an undirected graph $G = (V, E)$ with adjacency matrix A. It takes time $\Theta(1 + \deg(v))$ to compute $\deg(v)$ from A.	☐	☐
4	If every vertex of an undirected graph G has even degree, then G has an Eulerian walk.	☐	☐

1.) False. Consider



both $\langle A, B, C \rangle$ and $\langle B, A, C \rangle$ are valid topological orderings

2.) True. Let $V = \{v_1, v_2, \dots, v_n\}$. $\forall i \in \{1, \dots, n-1\}$, let the vertex v_i have $n-i$ outgoing edges, one to each vertex v_j, with $i < j \leq n$.
$\Rightarrow$ We have $(n-1) + (n-2) + \dots + (n-(n-1)) = \frac{(n-1) \cdot n}{2} = \binom{n}{2}$ edges.

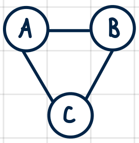
3.) False. Since we have an adjacency matrix we have to go through all entries in the respective row giving us a $\Theta(n)$ runtime.

4.) False. This is only true if the graph is connected

HS21

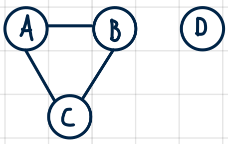
	Claim Kreis Zyklus	true	false
1	An undirected graph that contains a closed walk of even length always contains a cycle of even length as well.	☐	☐
	~~Let G be an undirected bipartite graph. Every cycle in G has even length.~~	☐	☐
	~~For all $n \in \mathbb{N}$ and all $0 \leq m \leq \frac{n(n-1)}{2}$, there exists a directed acyclic graph on n vertices with m edges.~~	☐	☐
2	An undirected graph $G = (V, E)$ with $\lvert E \rvert = \lvert V \rvert - 1$ is always connected.	☐	☐

1.) False. Consider



there exists a closed walk of even length $\langle A, B, C, A, B, C, A \rangle$ but no cycle of even length.

2.) False. Consider

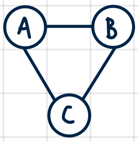


G is bipartite $\Leftrightarrow \exists A, B \subseteq V$ with $A \cap B = \emptyset$, $A \cup B = V$, such that for every edge $e = \{a, b\}$ $a \in A$ and $b \in B$

HS22

	Claim	true	false
1	Every graph that is connected and Eulerian is bipartite.	☐	☐
2	In a directed graph suppose there exists a walk with vertices s and t as endpoints. Then there exists a simple walk with vertices s and t as endpoints.	☐	☐
	~~In any tree $T = (V, E)$ with $\|V\| \geq 10$, we can always add at least one additional edge $e \notin E$ to T such that the resulting graph is bipartite (the set of vertices must remain the same)~~.	☐	☐
3	In every undirected graph $G = (V, E)$ with $\|V\| = \|E\| > 0$ there exists a simple cycle as a subgraph.	☐	☐
4	Given an undirected graph G with all degrees even, there always exists a way to direct the edges of G (i.e., convert each edge $\{a, b\}$ into either $a \to b$ or $b \to a$) such that in the resulting directed graph it holds that at every vertex v, the in-degree and out-degree are equal (despite different vertices can still have different in-degrees).	☐	☐

1.) False. Consider



2.) True. (try to prove this yourself, the proof is quite verbouse)

(to be exact: vertex disjoint)

3.) True. Since the graph is connected and $|E| > |V| - 1$, we don't have a tree and therefore there exist $u, v \in E$, with two disjoint u-v-paths. We can combine these to get a simple-cycle.

4.) True. To get this, find the Eulertour for every connected component. Let the directed edges point in the direction in which we traverse the Eulertour(s).
This is correct, since in an (undirected) Eulertour, everytime we "enter" a vertex we also "leave" it. Hence in the resulting graph it holds that $\forall v \in V$: $deg_{in}(v) = deg_{out}(v)$