ALGORITHMS & DATA STRUCTURES

November 21st

PLAN FOR TODAY

- Bouns Exericses
- Graph Theory Recap
- Quiz

EXERCISE 8.1

Exercise 8.1 *Party & Beer & Party & Beer* **(1 point).**

For your birthday, you organize a party and invite some friends over at your place. Some of your friends bring their partners, and it turns out that in the end everybody (including yourself) knows exactly 7 other people at the party (note that the relation of knowing someone is commutative, i.e. if you know someone then this person also knows you and vice versa). Show that there must be an even number of people at your party.

$G = (V, E)$ wo die Knoten sind die Leute

und $\{v_i, v_j\} \in E \iff$ i und j kennen sich

$$\boxed{\sum_{v \in V} \deg(v) = 2|E|}$$

$$\sum_{v \in V} \deg(v) = \sum_{v \in V} 7 = 7|V|$$

$7|V| = 2|E| \Rightarrow |V|$ gerade sein muss

EXERCISE 8.5

Theorem 1. *A graph is bipartite if and only if it does not contain any cycle of odd length.*

(i) Every graph G that is bipartite and Eulerian must have an even number of edges.

Wir nehmen an, $\exists$ G s.d. (1) G bipartit
(2) Eulersch
(3) Ungerade Anzahl von Kanten

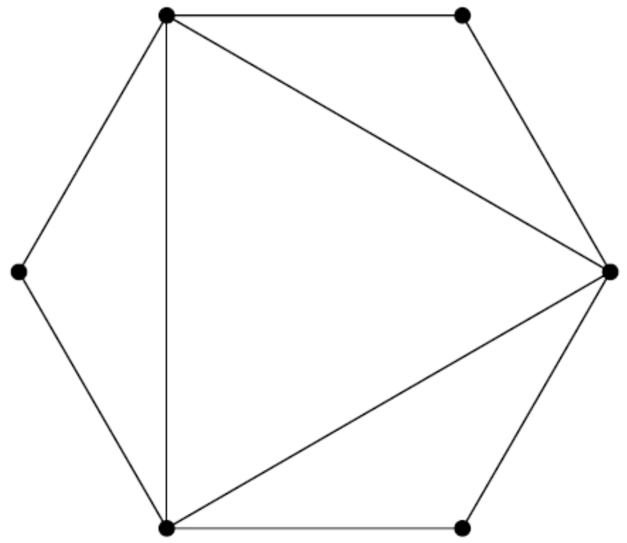
(2) + (3) : $\exists$ Kreis von Ungerade Länge

$\lightning$ zu Th. 1 (dieser Graph kann nicht bipartit sein)

(ii) Every Eulerian graph G that has an even number of vertices must also have an even number of edges.

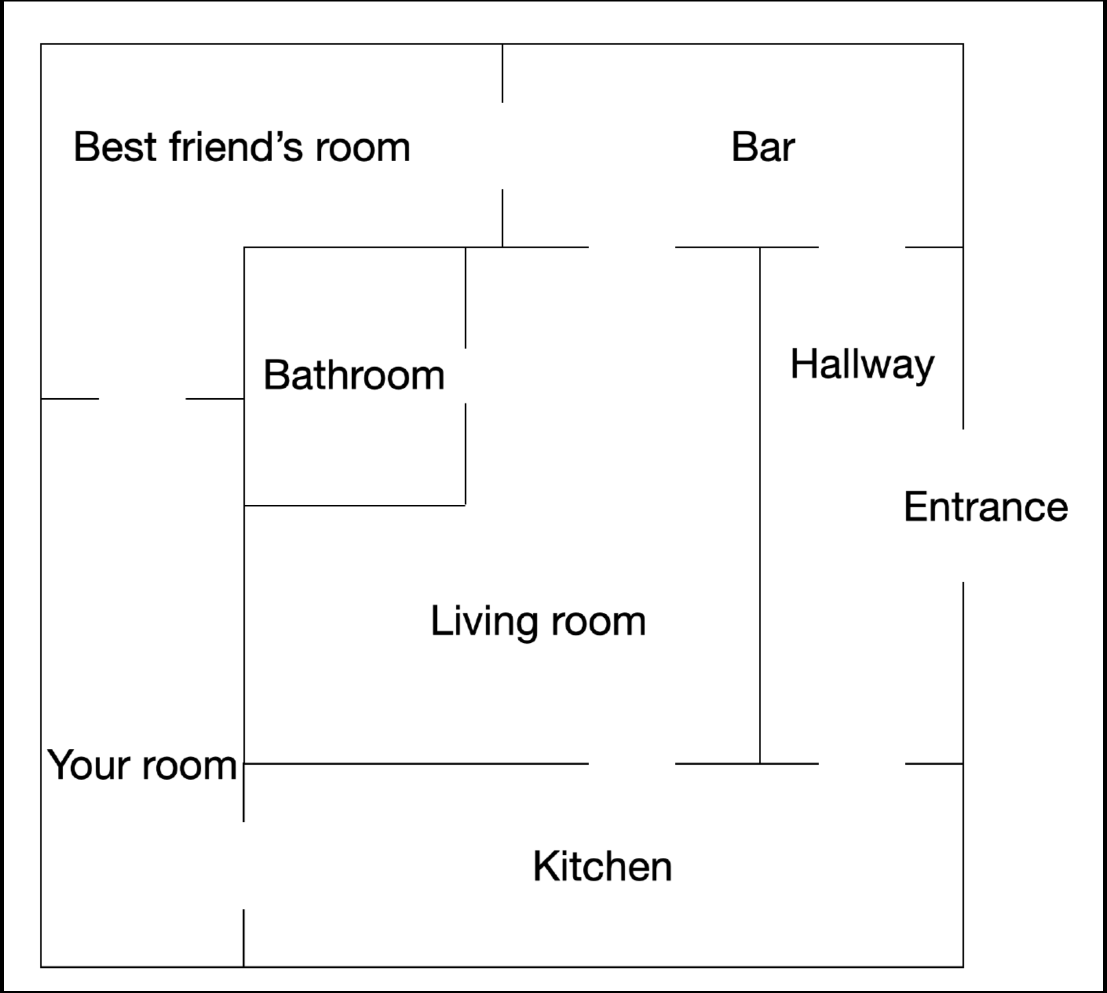
Solution:

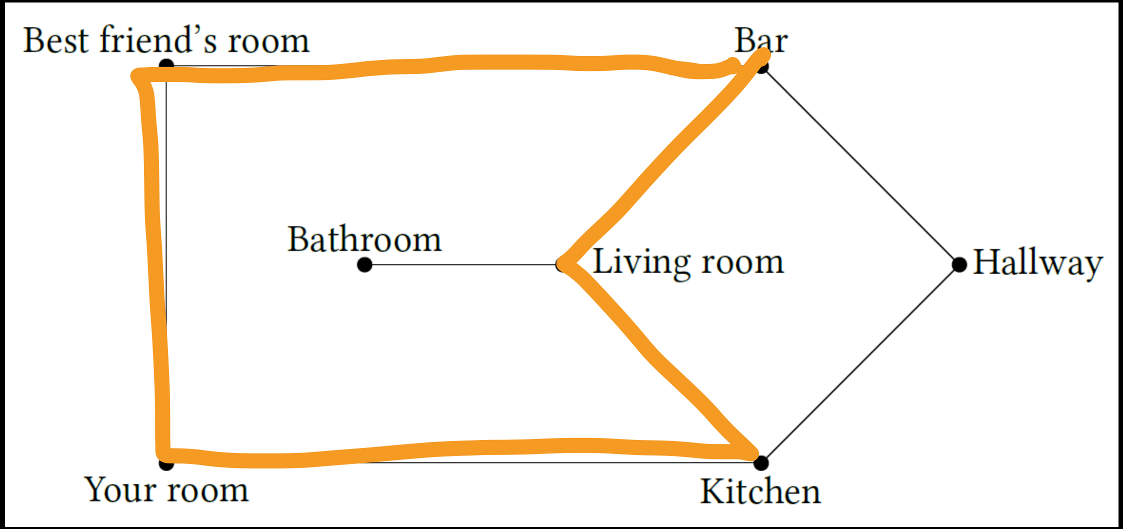
The following graph is a counterexample:



Indeed, the graph is clearly connected and one can easily check that the degree of each vertex is even, and hence the graph is Eulerian. However, it has an even number of vertices (6) but an odd number of edges (9).

b) You recently moved in with your best friend (see floor plan below) and you would like to repaint the room walls. Every room should be painted either in red or in purple (as these are your favorite colors), and you also would like that whenever you walk from a room to another room through a door, the color changes. Is that possible ?





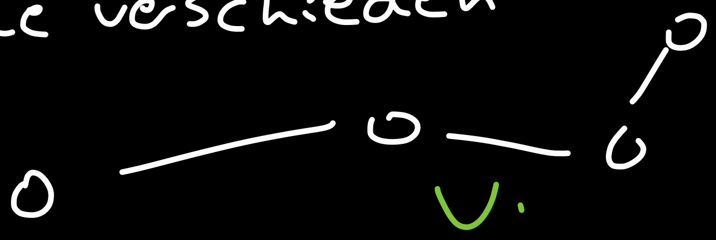
Ist G bipartit?

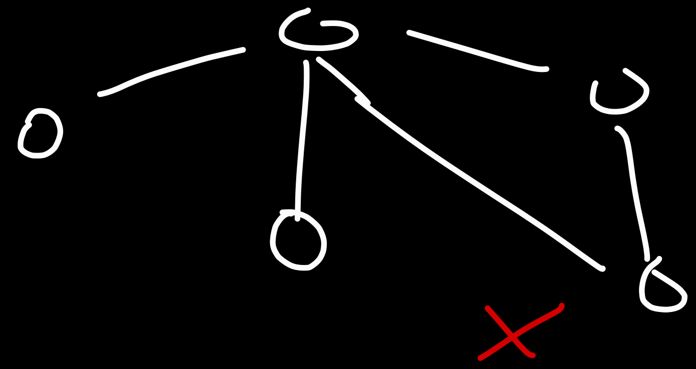
G nicht bipartit $\Rightarrow$ Unmöglich

GRAPH THEORY RECAP

Pfad

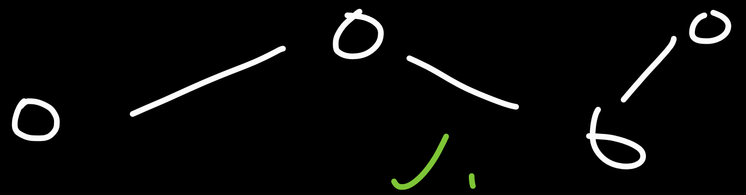
$v_1 \dots v_p$ s.d. $\{v_i, v_{i+1}\} \in E$ und $v_1 \dots v_p$ sind alle verschieden

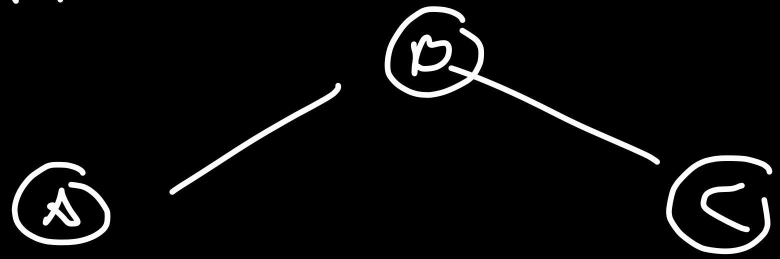




Weg

$v_1 \dots v_p$ s.d. $\{v_i, v_{i+1}\} \in E$





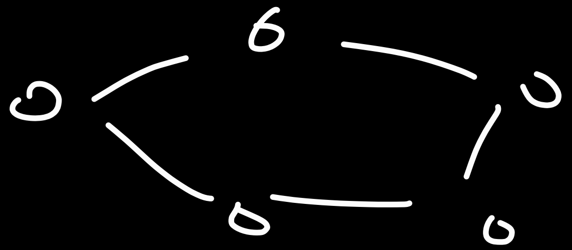
ABABABC

✓.

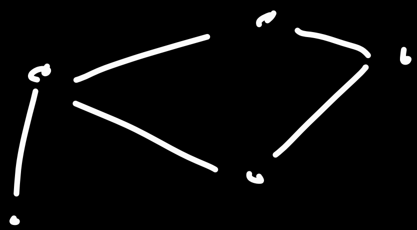
Weg $\Longleftarrow$ Pfad

$\not\Longrightarrow$

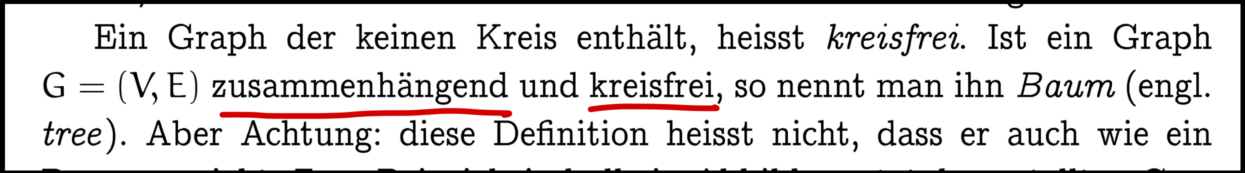
Kreis (cycle)



Zyklus



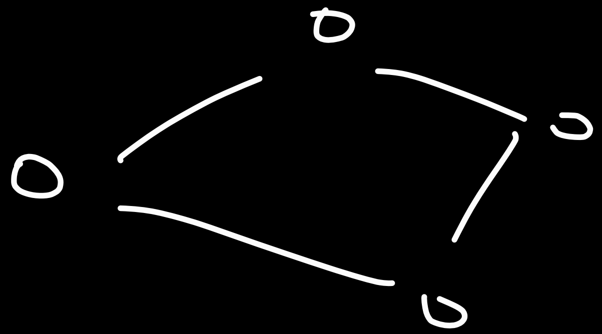
Baum



Ein Graph der keinen Kreis enthält, heisst *kreisfrei*. Ist ein Graph G = (V, E) zusammenhängend und kreisfrei, so nennt man ihn *Baum* (engl. *tree*). Aber Achtung: diese Definition heisst nicht, dass er auch wie ein

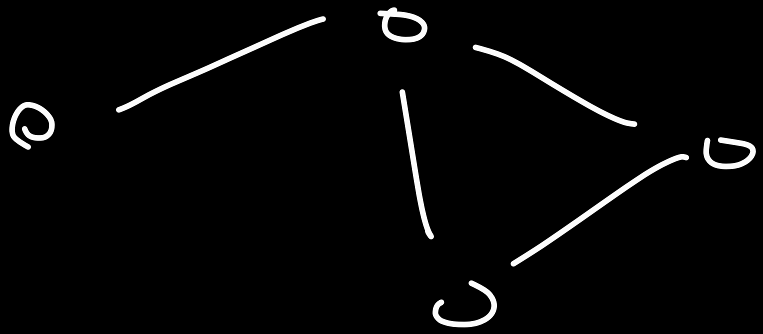
G ist ein Baum $\iff$ $|V| = |E| + 1$

Euler-tour (Eulerian circuit)



Alle Knoten haben geraden Grad

Euler wege



max. 2 Knoten mit ungeraden Grad

Hamiltonkreis

Kreis, die jeder Knoten genau einmal besucht (Ausnahme: Startknoten). Keine Polynomielle Algorithmen, unter der Annahme $P \neq NP$.

QUIZ

THE STACK IS A…

- FIFO data structure
- LIFO data structure

THE STACK IS A...

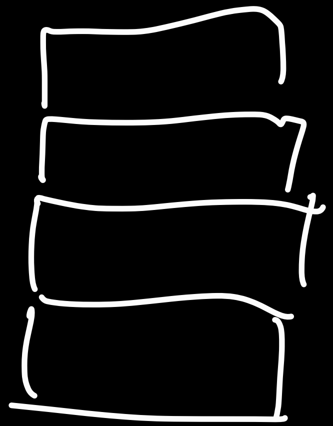
Queue

- FIFO data structure
- LIFO data structure

PUSH O(1)

POP O(1)

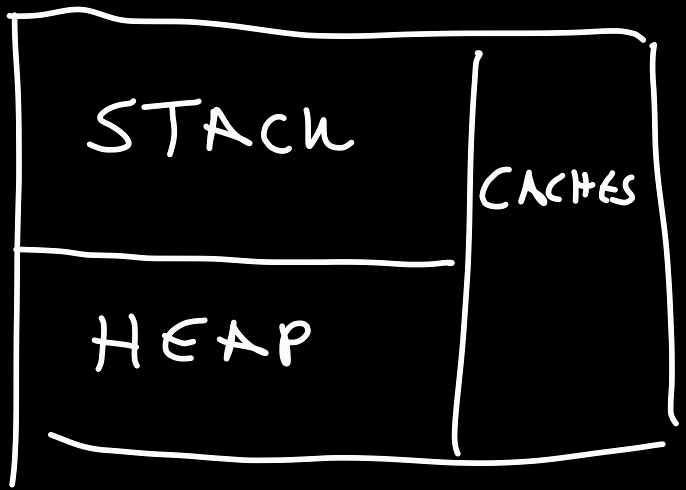
[TOP] O(1)



Wie können wir einen Stack implementieren?

Liste

Das ist nicht der Stack von Stack-overflow



WHAT IS THE DIFFERENCE BETWEEN ABSTRACT DATA TYPE AND DATA STRUCTURE?

ADT Eine liste von Operationen, die erlaubt sind

INSERT, DELETE, SEARCH

Datenstruktur Eine Implementation von de ADT

liste, ..., AVL Bäume

z.B Die selbe ADT kann auf verschiedene weise (mit verschiedene Komplexität) implementiert werden.

WHAT OF THE FOLLOWING IS THE MOST EFFICIENT WAY TO IMPLEMENT A PRIORITY QUEUE?

- Sorted array
- Binary search tree
- AVL tree
- Heap

WHAT OF THE FOLLOWING IS THE MOST EFFICIENT WAY TO IMPLEMENT A PRIORITY QUEUE?

- Sorted array
- Binary search tree
- AVL tree
- Heap

In einem Algo benutzen wir ADTs und wir sollen denn die effizienteste Datenstruktur wählen, die den nötigen ADT (unter unseren Annahmen) implementiert.

YOU CAN IMPLEMENT A DATA STRUCTURE FOR DICTIONARY WITH A HEAP

- True
- False

YOU CAN IMPLEMENT A DATA STRUCTURE FOR DICTIONARY WITH A HEAP

- True
- False

INSERT
DELETE
SEARCH

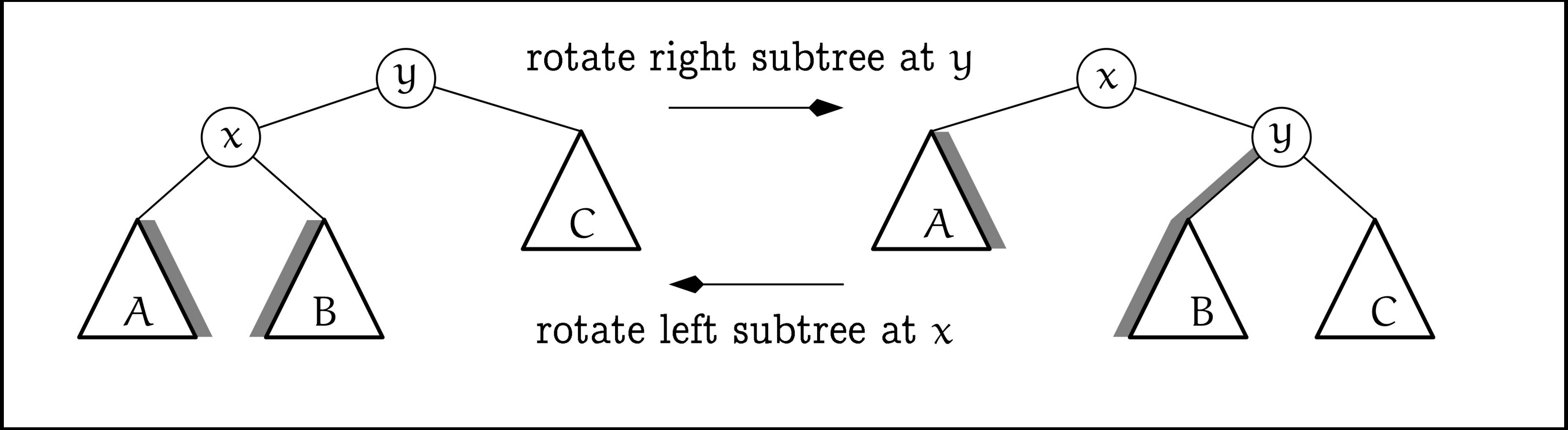
SHORT SUMMARY OF AVL TREES

Ihr kennt binary search trees

$$\left.\begin{array}{l}\text{INSERT}\\ \text{DELETE}\\ \text{SEARCH}\end{array}\right\} O(h)$$

AVL Bäume benutzen <u>Rotationen</u>, s.d. wir sicher sind dass $h \in O(\log n)$. <u>AVL Property:</u> für jeden Knoten, die Höhe nach links und nach rechts unterschieden sich ≤ 1

SHORT SUMMARY OF AVL TREES


y
x
A
B
C
rotate right subtree at y
rotate left subtree at x
x
y
A
B
C