

## GDP Klausur 26.02.2020 (Gedächtnisprotokoll)

### Graphen

3 Merkmale von Modellen nennen

-Definition gerichteter Graph geben

Gegeben

G gerichteter Graph

$V = \{p, q, r, s, t\}$

Adjazenzmatrix =

0 1 1 0 0

0 0 0 1 1

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1

# ich bin mir tatsächlich relativ sicher, dass das die richtige Matrix ist, weil ich mich an meine Zeichnung erinnern konnte, jedenfalls hatte s auf jeden Fall eine Schlinge

-Graph G zeichnen

- Adjazenzliste angeben

- Kreise auf G nennen

- Zykel, der kein Kreis ist nennen

- Wie wird durch die Knoten durchlaufen, BFS und DFS

## Prozedural

```
def f(x):  
    x = x+2  
    return g(y)  
  
def g(x):  
    y = x**2  
    return y // 5  
  
y = 2  
x = f(y)  
print(x)
```

-Nenne den formalen und aktuellen Parameter

- Gib an, was print ausgibt

- Zeichne Stackframe, der existiert, genau wenn g aufgerufen wird (nur ein Stack)

- Achte darauf, welche Variable zu welchem Zeitpunkt bekannt ist.

- Gib Algorithmus an, der berechnet, ob die Summe der ersten Hälfte der Elemente von einer Liste gleich der Summe der 2. Hälfte der Elemente ist (Python oder Pseudocode)

#glaube, da war noch ein Programm, aber ich erinnere mich nicht mehr daran

## Laufzeitanalyse

Es war wieder sowas zum ankreuzen, aber an die erinnere ich mich leider nicht  
ich weiß nur, die letzte war mit log

eine Funktion war angeben, wo man zeigen musste dass sie  $\Theta(n^3)$  ist

und dann war noch eine Aufgabe mit Algorithmus A und B,

ich glaube  $T_a$  war Element von  $\Omega(n^3)$ , und  $T_b$  war Element von  $O(n^2)$  oder so

und man musste wieder ankreuzen, welche Schlussfolgerungen man daraus ziehen kann

also  $T_a$  element von  $\Theta(T_b)$  und umgekehrt und alle anderen Konstellationen

ohne Begründung

und dann musste man argumentieren, welcher Algorithmus zeiteffizienter ist

## OOP

```
class Hamster:
    age = 0 #vollendete Lebensjahre
def __init__(self,age):
    self.age = age
def Hurra(self):
```

- Welche Datenelemente hat die Klasse Hamster?
- Funktion Hurra definieren, die so oft Hurra ausgibt, wie der Hamster Lebensjahre vollendet hat (benutze dafür print())
- Lege Objekt myHamster an, mit dem Alter 2 (ohne Klasse zu verändern)

## Funktional

Nenne 2 Merkmale funktionaler Programmierung

```
from functools import reduce
def prolong (L, n):
    if n == 0:
        return L
    L.append (10*L[len(L) -1] )#bin mir nicht mehr sicher, wie das genau hier war
    return prolong(L, n-1)      # aber es wurde jedenfalls immer mal 10, also 30,300,3000 usw.

L= [3]
print(reduce(lambda x,y: x-y, L, 100))
n = input()
prolong(L,n)

K = L # nur hier verändern
```

was gibt das erste print aus?

Was ist die Liste, nachdem prolong() aufgerufen wurde, wenn n = 5 ist  
verändern sie K so, dass

- i) Für jedes Element der Liste 3 ausgegeben wird
- ii) Nur Listenelemente die kleiner als 10.000 sind ausgegeben werden

Denken Sie hierbei funktional (#also map und filter)

Schreibe Liste [2,2,4,1,1] (oder so) als Links und als Rechtssequenz

Zur Erinnerung : rest und first für Linkssequenzen werden erklärt

definiere Funktion append, die für eine LS xs und ein Element x, eine LS ausgibt, an die x als neues  
letztes Element angehängt ist

**Nächste Aufgabe** (ich weiß die Überschrift nicht mehr)

- Compiler im Gegensatz zu Interpreter erklären
- Eine Programmiersprache, die (reine) Interpretersprache ist
- Eine Programmiersprache, die (reine) Compilersprache ist
- einen möglichen Syntaxfehler in Python angeben
- Ein Fehler in Python, der kein Syntaxfehler ist angeben

Schreibe in AASS ein Programm das gleichwertig ist zu:

```
while x > 0:
```

```
    x+= 3
```

```
y = y*x
```

Dabei gilt:

$$\begin{array}{r} x y \\ \hline 4 8 \end{array}$$

## Letzte Aufgabe

Eine nicht berechenbare Funktion angeben (Eingabe/Ausgabe)

eine berechenbare Funktion angeben (Eingabe, Ausgabe)

Definition von abzählbar unendlich geben