

---

# Datenstrukturen und Effiziente Algorithmen

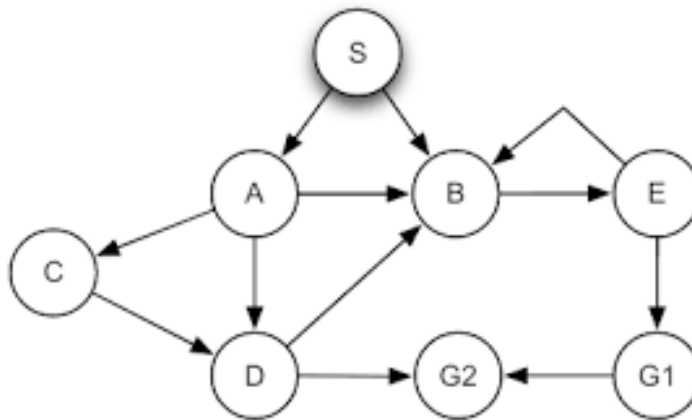
## Wintersemester 2019

---

### Präsenzaufgaben 1

#### Aufgabe 1. (Wdh Tiefensuche)

A Gib eine Adjazenzlisten-Repräsentation des folgenden Graphen  $G = (V, E)$  an.



B Auf Basis dieser Repräsentation, gib die discover- und finish-times aller Knoten an, startend mit  $S$ .

C Gib eine möglichst kleine Menge von Kanten  $E' \subset E$  an, s.d.  $G' = (V, E \setminus E')$  kreisfrei ist. Gib eine topologische Ordnung der Knoten des kreisfreien Graphen an.

#### Aufgabe 2. (O-Notation)

A Beweise, dass  $2n^2 + 3n + 1 \in O(n^2)$ .

B Ist  $2^{n+1} \in O(2^n)$ ?

C Ist  $2^{2n} \in O(2^n)$ ?

#### Aufgabe 3. (Menge der Tiefensuche Kanten ist Wald)

Beweise die Aussage aus der Vorlesung:  $(V, E_{pred})$  ist azyklisch.

#### Aufgabe 4. (Python)

Melde dich auf [Jupyter Uni Greifswald](#) mit deiner Uni-Kennung an.

Kopiere das Notebook **DFS experiments.ipynb** von Moodle in dein Verzeichnis und öffne es.

Wir gehen das Notebook zusammen durch. Dann wollen wir folgendes Problem lösen:

Finde einen effizienten Algorithmus, der die stark zusammenhängenden Komponenten eines gerichteten Graphen ausgibt.

Ein gerichteter Graph ist stark zusammenhängend, wenn zwischen allen geordneten Knotenpaaren ein Pfad existiert. Die stark zusammenhängenden Komponenten eines Graphen sind eine Partition der Knoten, sodass jede Teilmenge einen stark zusammenhängenden Teilgraphen darstellt und maximal ist.

*Tipp: Benutze DFS (ggf. mehrmals).*

#### Aufgabe 4. (C++)

Schreibe ein einfaches Programm, das für den Graphen aus Aufgabe 1 bei Angabe eines Knotens (zB ./meinProgramm 'A') alle zu diesem Knoten adjazenten Knoten ausgibt.

*Tipp: Benutze den Template Code von Übungsblatt 1.*