

## PAD Woche 8: Graphalgorithmen II

**Inhalte: elementare Datenstrukturen, Stack, Queue, Breitensuche**

### Datenstrukturen:

- Eine Datenstruktur beschreibt eine Methode zum Speichern/Organisieren von Daten/Information
- Datenstrukturen stellen üblicherweise Operationen zur Verfügung, um Information hinzuzufügen und zu finden

### Stack:

- Ein Stack ist eine Datenstruktur, die eine Multimenge von Elementen speichert
- *push*: Füge ein Element zum Stack hinzu
- *pop*: Entferne jenes Element aus dem Stack, das zuletzt mittels push hinzugefügt wurde (und gib es zurück)
- Visualisierung: Tellerstapel
- push nach oben, pop von oben
- LIFO (last in first out)

### Queue:

- Eine Queue ist eine Datenstruktur, die eine Multimenge von Elementen speichert
- *enqueue*: Füge ein Element zur Queue hinzu
- *dequeue*: Entferne jenes Element aus der Queue, das zuerst mittels enqueue hinzugefügt wurde (und gib es zurück)
- Visualisierung: Warteschlange an der Kasse
- enqueue am Ende, dequeue am Anfang
- FIFO (first in first out)

### Aufgabe 1 (10 min):

- Führen Sie die folgenden Operationen für einen anfangs leeren Stack aus;  
push('E') push('A') push('S') pop() push('Y') pop() push('Q') push('U')  
push('E') pop() pop() pop() push('S') push('T') pop() pop() pop()  
push('I') push('O') pop() push('N') pop() pop() pop()
- Führen Sie die folgenden Operationen für eine anfangs leere Queue aus;  
enq('E') enq('A') enq('S') deq() enq('Y') deq() enq('Q') enq('U') enq('E')  
deq() deq() deq() enq('S') enq('T') deq() deq() deq() enq('I') enq('O')  
deq() enq('N') deq() deq() deq()

### Tiefensuche (Depth-First-Search) Algorithmus (für Startknoten s und Zielknoten t):

push s

Wiederhole die folgenden Anweisungen aus bis der Stack leer ist:

pop x

Falls x noch nicht markiert wurde, führe die folgenden Anweisungen aus:

Markiere x als 'besucht'

Falls x der Zielknoten t ist, berichte, dass das Ziel gefunden wurde

Andernfalls, gehe alle Kanten, die x als Endpunkt haben durch und

führe für jede solche Kante push y aus, wobei y jeweils den anderen Endpunkt der Kante bezeichnet

### Aufgabe 2 (20 min):

Im Tiefensuche-Algorithmus haben wir einen Stack verwendet. Betrachten wir nun eine Variante des Algorithmus, in dem wir den Stack durch eine Queue ersetzen. In welcher Reihenfolge werden die Knoten des Graphs mit dieser Modifikation besucht?

### Breitensuche (Breadth-First-Search) Algorithmus (für Startknoten x und Zielknoten t):

enqueue s

Wiederhole die folgenden Anweisungen aus bis die Queue leer ist:

    dequeue x

    Falls x noch nicht markiert wurde, führe die folgenden Anweisungen aus:

        Markiere x als 'besucht'

        Falls x der Zielknoten t ist, berichte, dass das Ziel gefunden wurde

        Andernfalls, gehe alle Kanten, die x als Endpunkt haben durch und

        führe für jede solche Kante enqueue y aus, wobei y jeweils den anderen Endpunkt der Kante bezeichnet

### Distanz-Werte:

- Der Breitensuche-Algorithmus kann so erweitert werden, dass zusätzlich ein Distanz-Wert für jeden Knoten berechnet wird.
- $\text{dist}(s) = 0$  (Wenn enqueue s ausgeführt wird)
- $\text{dist}(y) = \text{dist}(x) + 1$  (Wenn enqueue y ausgeführt wird)
- Dadurch wird die Distanz von s zu jedem Knoten x berechnet, d.h., die Länge des kürzesten Wegs von s nach x.