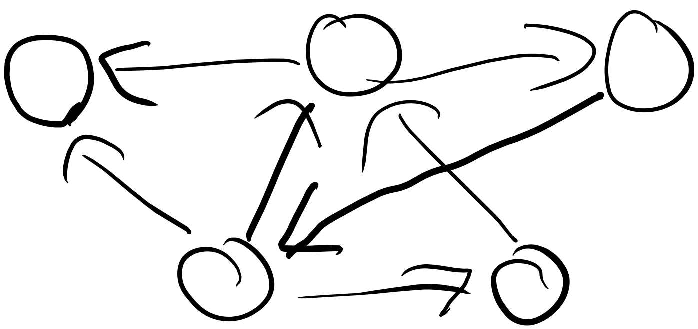
Grundzüge der Informatik 1

Vorlesung 19 - flipped classroom

Graphalgorithmen

Definition (gerichteter Graph)

- Ein *gerichteter Graph* ist ein Paar (V,E), wobei V eine endliche Menge ist und $E \subseteq V \times V$.
- V heißt Knotenmenge des Graphen
- Die Elemente aus V sind die Knoten des Graphen
- E heißt Kantenmenge des Graphen
- Die Elemente aus E sind die Kanten des Graphen



Graphalgorithmen

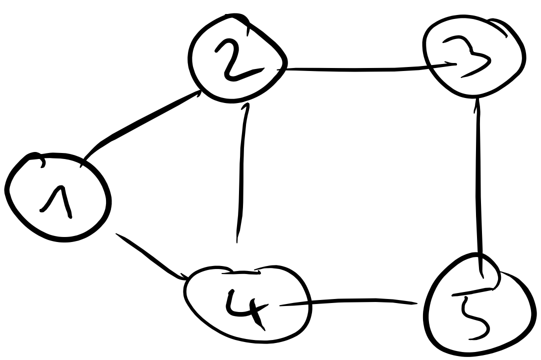
Definition (ungerichteter Graph)

- Ein *ungerichteter Graph* ist ein Paar (V,E), wobei V eine endliche Menge ist und E Teilmenge der Menge aller Paare von Elementen aus V ist
- V heißt Knotenmenge des Graphen
- Die Elemente aus V sind die Knoten des Graphen
- E heißt Kantenmenge des Graphen
- Die Elemente aus E sind die Kanten des Graphen
- Wir stellen Kanten aus V wie im gerichteten Fall durch (u,v) dar und nehmen an, dass die Kante (u,v) gleich der Kante (v,u) ist
- Manchmal repräsentieren wir einen ungerichteten Graph durch einen gerichteten, indem wir jede Kante (u,v) durch die gerichteten Kanten (u,v) und (v,u) ersetzen

Graphalgorithmen

Adjazenzmatrixdarstellung

- Knoten sind nummeriert von 1 bis $|V|$
- $|V|\times|V|$ Matrix $A = (a_{ij})$ mit
- $a_{ij} = 1$, wenn $(i,j)\in E$ und $a_{ij} = 0$, sonst
- Bei ungerichteten Graphen gilt $A = A^T$

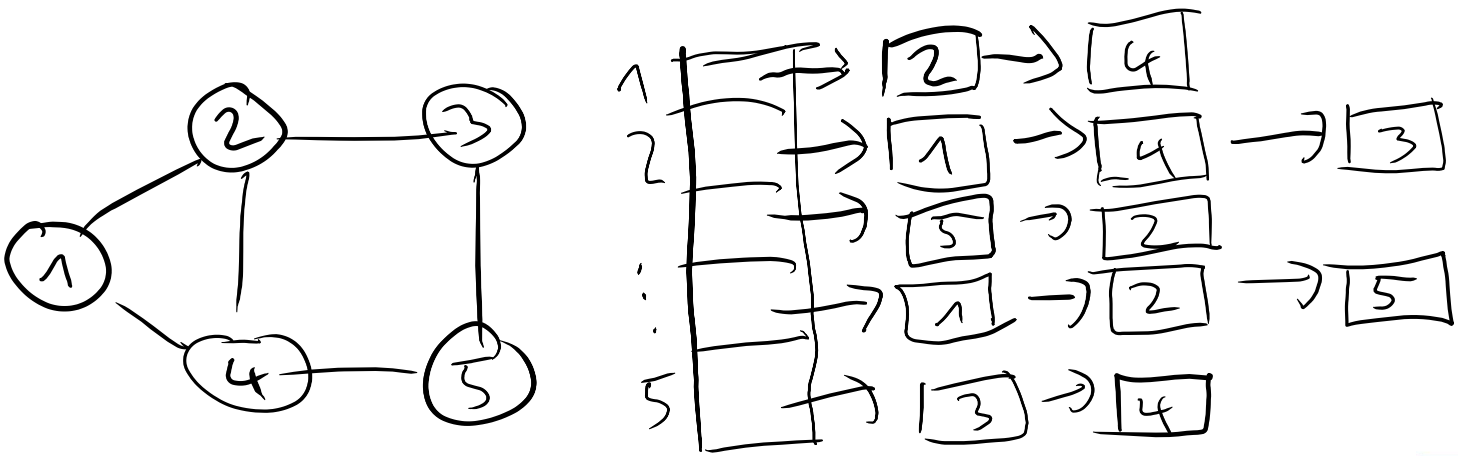


$$
\begin{array}{c|ccccc}
 & 1 & 2 & \cdots & & 5 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 \\
\vdots & 1 & 0 & 1 & 1 & 0 \\
 & 0 & 1 & 0 & 0 & 1 \\
 & 1 & 1 & 0 & 0 & 1 \\
5 & 0 & 0 & 1 & 1 & 0
\end{array}
$$

Graphenalgorithmen

Adjazenzlistendarstellung

- Feld Adj mit |V| Listen (eine pro Knoten)
- Für Knoten v enthält Adj[v] eine Liste aller Knoten u mit $(v,u)\in E$
- Die Knoten in Adj[v] heißen zu v benachbart
- Ist G ungerichtet, so gilt: $v\in Adj[u] \Leftrightarrow u\in Adj[v]$



Datenstrukturen für Graphen

Aufgabe 1

- Die Transposition eines gerichteten Graph $G=(V,E)$ ist der Graph $G^T=(V,E^T)$, wobei $E^T = \{(v,u) \in V \times V : (u,v) \in E\}$. Entwickeln Sie einen Algorithmus, der G^T für einen Eingabegraph G in Adjazenzlistendarstellung berechnet. Sie können dabei annehmen, dass $V = \{1,\ldots,n\}$ ist.
- Was ist die Laufzeit Ihres Algorithmus?

Datenstrukturen für Graphen

Transposition(G)

1. **new array** AdjT[1…|V|]
2. **for each** v∈V **do**
3.   **for each** u∈Adj[v] **do**
4.   Einfügen(AdjT[u],v)

Datenstrukturen für Graphen

Transposition(G)

1. **new array** AdjT[1…|V|] $O(n)$
2. **for each** $v \in V$ **do** $O(n)$
3. **for each** $u \in Adj[v]$ **do** $O(\Sigma\ (1+outdeg(v)))$
4. Einfügen(AdjT[u],v) $O(\Sigma\ (1+outdeg(v)))$

- Da die Summe der Knotengrade der Anzahl der Kanten entspricht, ist die Laufzeit $O(|V|+|E|)$

Datenstrukturen

Aufgabe 2

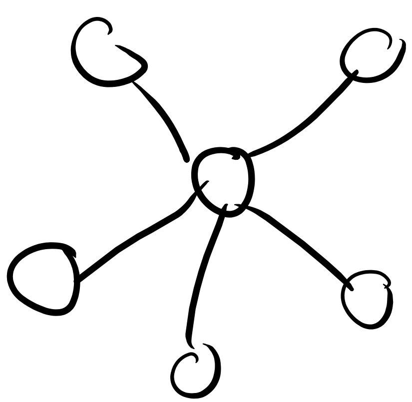
- Das Quadrat eines ungerichteten Graph $G=(V,E)$ ist der Graph $G^{(2)}=(V,E^{(2)})$, wobei $(u,w)\in E^{(2)}$, gdw. es einen Knoten v gibt, so dass $(u,v)\in E$ und $(v,w)\in E$.
- Zeigen Sie: Es gibt Graphen mit $O(n)$ Kanten, deren Quadrat $\Omega(n^2)$ Kanten haben

Datenstrukturen

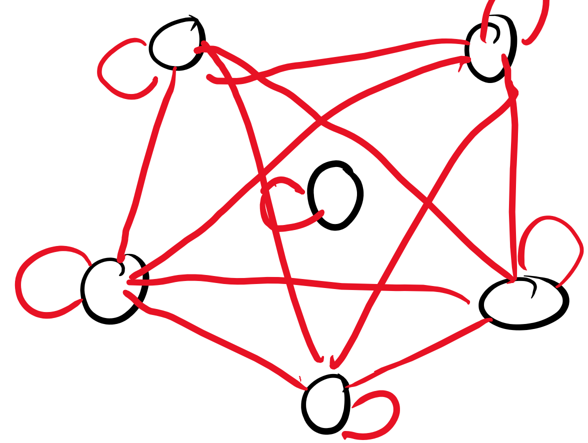
Aufgabe 2

- Für jedes n erfüllt der Sterngraph (ein Knoten, der mit n-1 anderen Knoten verbunden ist) die Anforderungen
- Beispiel (n=6):

G



$G^{(2)}$



Datenstrukturen

Aufgabe 3

- Das Quadrat eines ungerichteten Graph $G=(V,E)$ ist der Graph $G^{(2)}=(V,E^{(2)})$, wobei $(u,w)\in E^{(2)}$, gdw. es einen Knoten v gibt, so dass $(u,v)\in E$ und $(v,w)\in E$.
- Entwickeln Sie einen Algorithmus, der das Quadrat eines Graphen $G=(V,E)$ berechnet, der in Adjazenzmatrixdarstellung abgespeichert ist
- Was ist die Laufzeit Ihres Algorithmus?

Datenstrukturen für Graphen

Quadrat(G) * Sei A die Adjazenzmatrix von G

1. **new array** B[1…n][1…n]
2. **for** i=1 **to** n **do**
3.   **for** j=1 **to** n **do**
4.   **for** k=1 **to** n **do**
5.   **if** A[i][j]=1 und A[j][k]=1 **then** B[i][k]=1

Datenstrukturen für Graphen

Quadrat(G)		* Sei A die Adjazenzmatrix von G
1.	**new array** B[1…n][1…n]	$O(n^2)$
2.	**for** i=1 **to** n **do**	$O(n)$
3.	**for** j=1 **to** n **do**	$O(n^2)$
4.	**for** k=1 **to** n **do**	$O(n^3)$
5.	**if** A[i][j]=1 und A[j][k]=1 **then** B[i][k]=1	$O(n^3)$