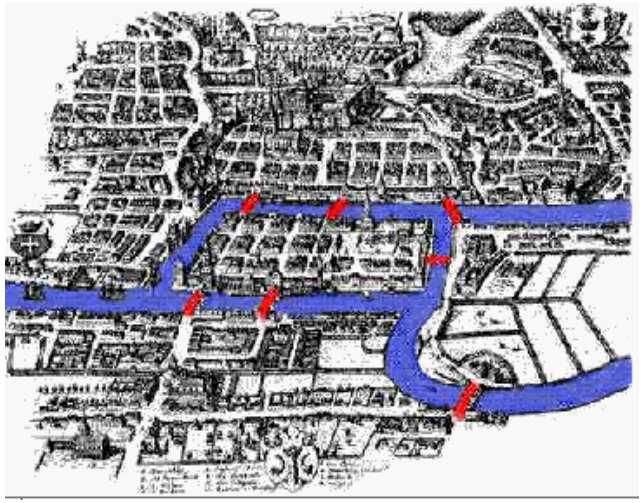
Kap. 8: Repräsentation von Graphen

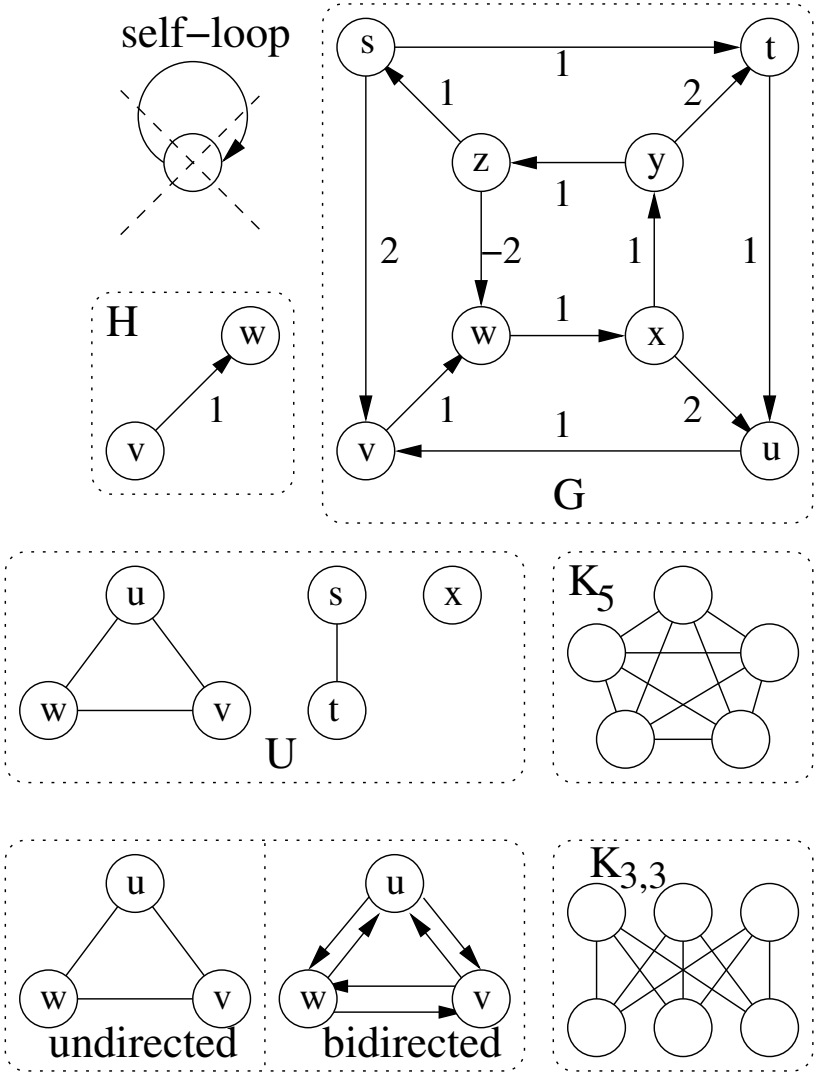
Einleitung

- 1736 stellt L. Euler die folgende "touristische" Frage:
- Straßen- oder Computernetzwerke
- Zugverbindungen (Raum und Zeit)
- Soziale Netzwerke (Freundschafts-, Zitier-, Empfehlungs-,...)
- Aufgabenabhängigkeiten $\rightsquigarrow$ Scheduling-Probleme
- Werte und arithmetische Operationen $\rightsquigarrow$ Compilerbau
- ...



Repräsentation von Graphen

- Was zählt, sind die Operationen!
- Eine triviale Repräsentation:
- Felder
- Verkettete Listen
- Matrizen
- Implizit
- Diskussion



Notation und Konventionen

- Graph $G = (\underbrace{V}_{\text{Knoten}}, \underbrace{E}_{\text{Kanten}})$:
- $n = |V|$
- $m = |E|$
- Knoten: s, t, u, v, w, x, y, z
- Kanten $e \in E$.
  Oder: Knotenpaare (manchmal Knotenmengen der Größe 2)

Notation und Konventionen

- Graph $G = (\underbrace{V}_{\text{Knoten}} , \underbrace{E}_{\text{Kanten}})$:
- $n = |V|$
- $m = |E|$
- Knoten: s, t, u, v, w, x, y, z
- Kanten $e \in E$.
  Oder: Knotenpaare (manchmal Knotenmengen der Größe 2)

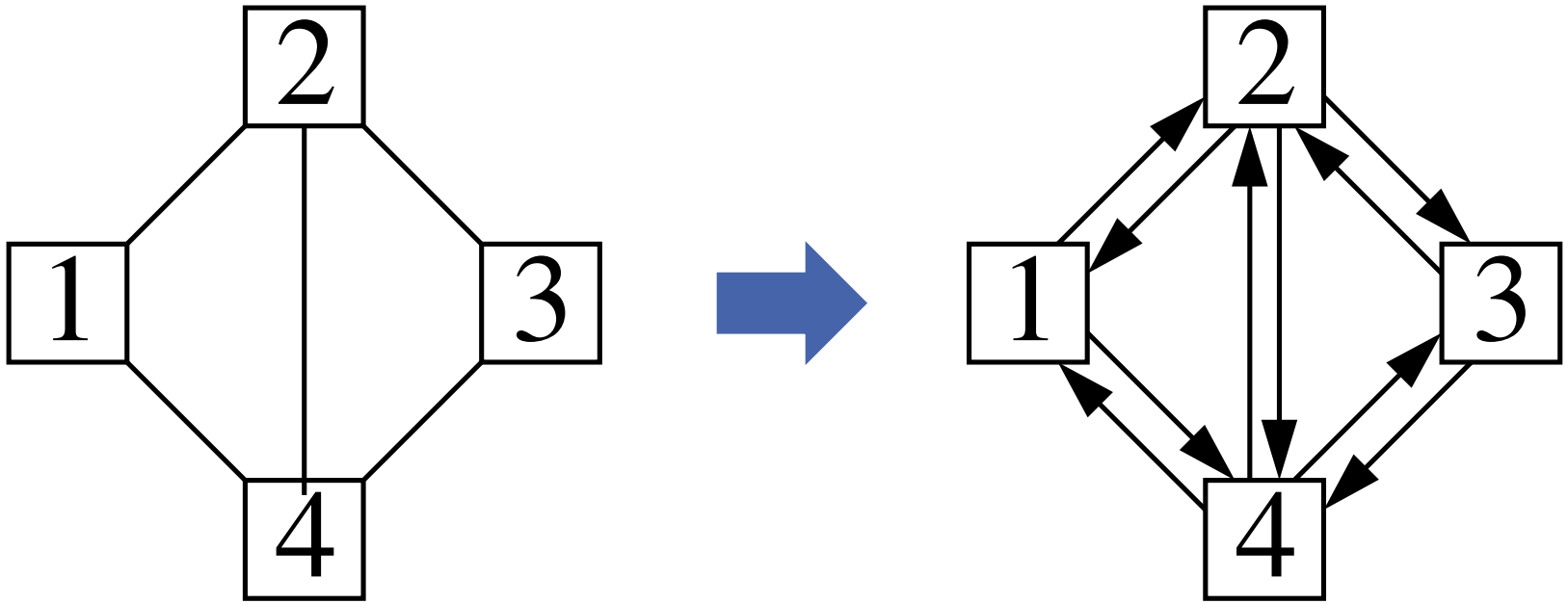
WICHTIG: Buchstabenzuordnungen sind unverbindliche Konvention

- Manchmal werden ganz andere Buchstaben verwendet.
- Im Zweifel immer genau sagen, was was ist.

Das gilt für die ganze theoretische Informatik!

Ungerichtete $\longrightarrow$ gerichtete Graphen

Meist repräsentieren wir
ungerichtete Graphen durch doppelt gerichtete Graphen
$\rightsquigarrow$ wir konzentrieren uns auf gerichtete Graphen



Operationen

Ziel: O(Ausgabegröße) für alle Operationen

Grundoperationen:

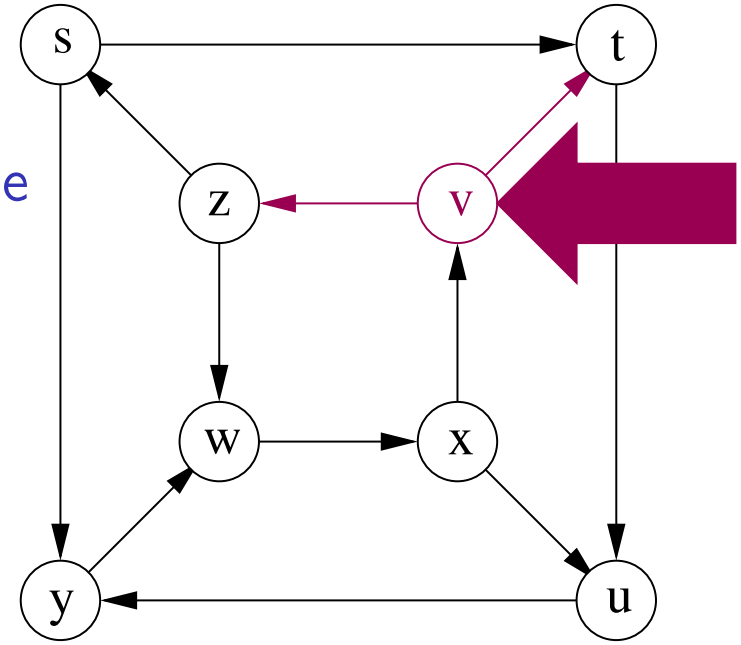
- Statische Graphen:

  Konstruktion, Konversion und Ausgabe ($O(m+n)$ Zeit)

  Navigation: Gegeben v, finde ausgehende Kanten.

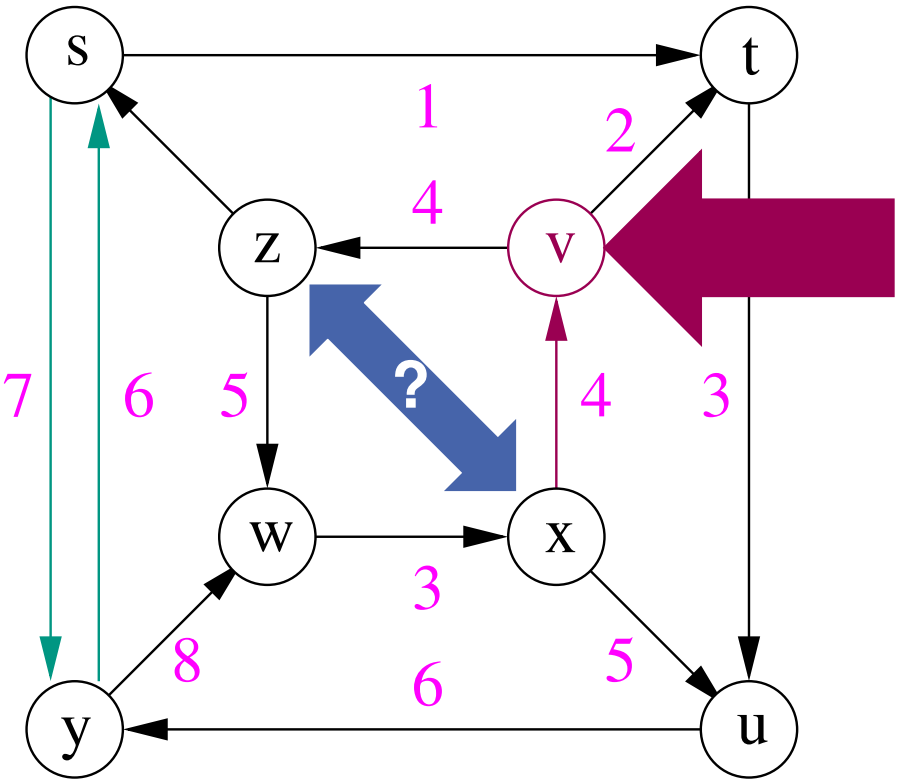
- Dynamische Graphen:

  Knoten/Kanten einfügen/löschen



Weitere Operationen

- Zugriff auf assoziierte Information
- Mehr Navigation: Finde eingehende Kanten
- Kantenanfragen: $(z, x) \in E$?

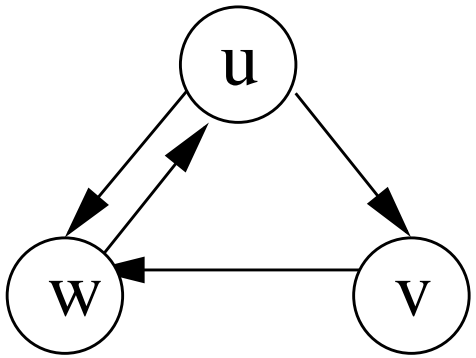


Kantenfolgenrepräsentation

Folge von Knotenpaaren (oder Tripel mit Kantengewicht)

- \+ kompakt
- \+ gut für I/O
- – Fast keine nützlichen Operationen – außer alle Kanten zu durchlaufen

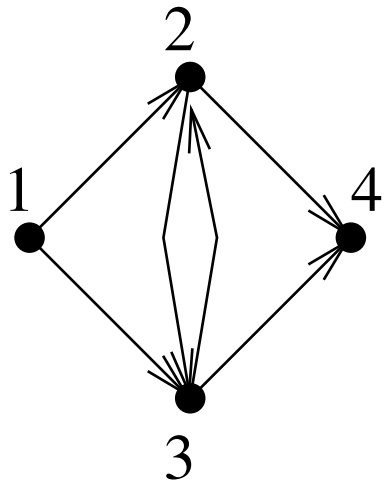
Beispiele: Übung: isolierte Knoten suchen, Kruskals MST-Algorithmus (später), Konvertierung.

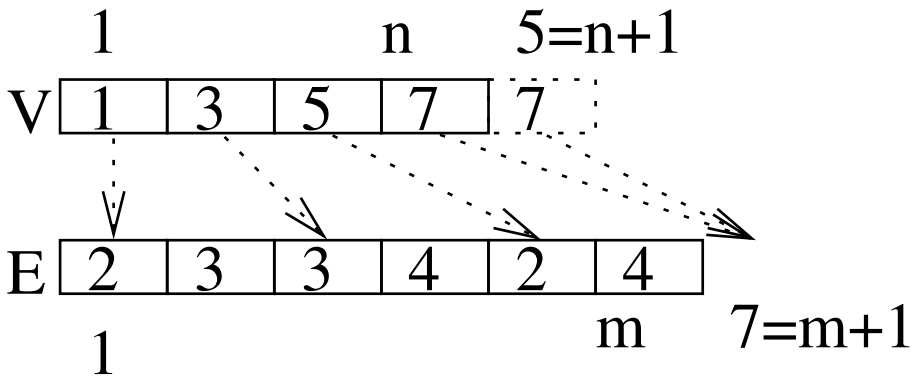


$$\Leftrightarrow \langle (u,v),(v,w),(w,u),(u,w) \rangle$$

Adjazenzfelder

- $V = 1..n$ oder $0..n-1$
- Kantenfeld `E` speichert Ziele und zwar gruppiert nach Startknoten
- `V` speichert Index der ersten ausgehenden Kante
- Dummy-Eintrag $V[n+1]$ speichert $m+1$





Beispiel: Ausgangsgrad$(v) = \mathtt{V}[v+1] - \mathtt{V}[v]$

Kantenliste $\rightarrow$ Adjazenzfeld

Zur Erinnerung: KSort (BucketSort)

Function adjacencyArray(EdgeList)

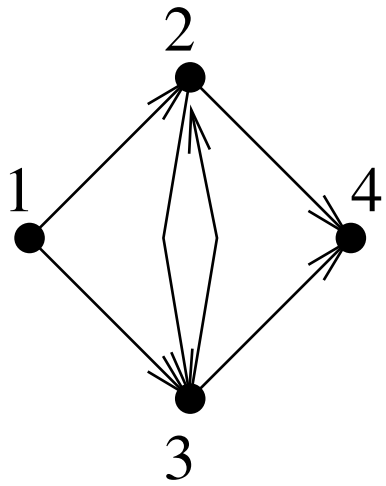
V$=\langle 1,0,\ldots,0\rangle$: **Array** $[1..n+1]$ **of** $\mathbb{N}$

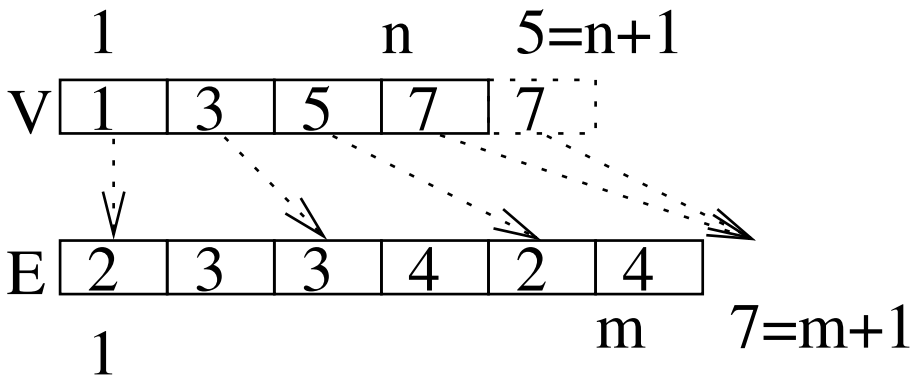
foreach $(u,v) \in$ EdgeList **do** $V[u]{+}{+}$ // count

for $v := 2$ **to** $n+1$ **do** $V[v] \mathrel{+}= V[v-1]$ // prefix sums

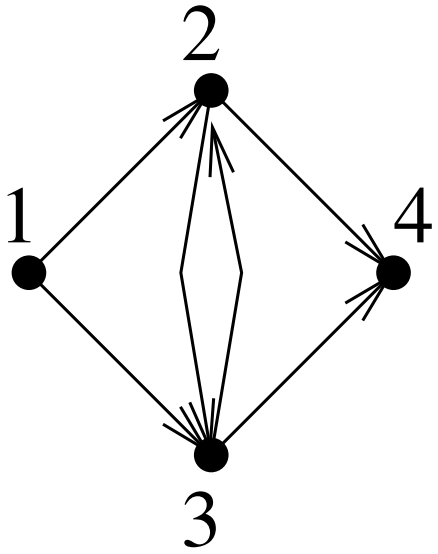
foreach $(u,v) \in$ EdgeList **do** $E[--V[u]] = v$ // place

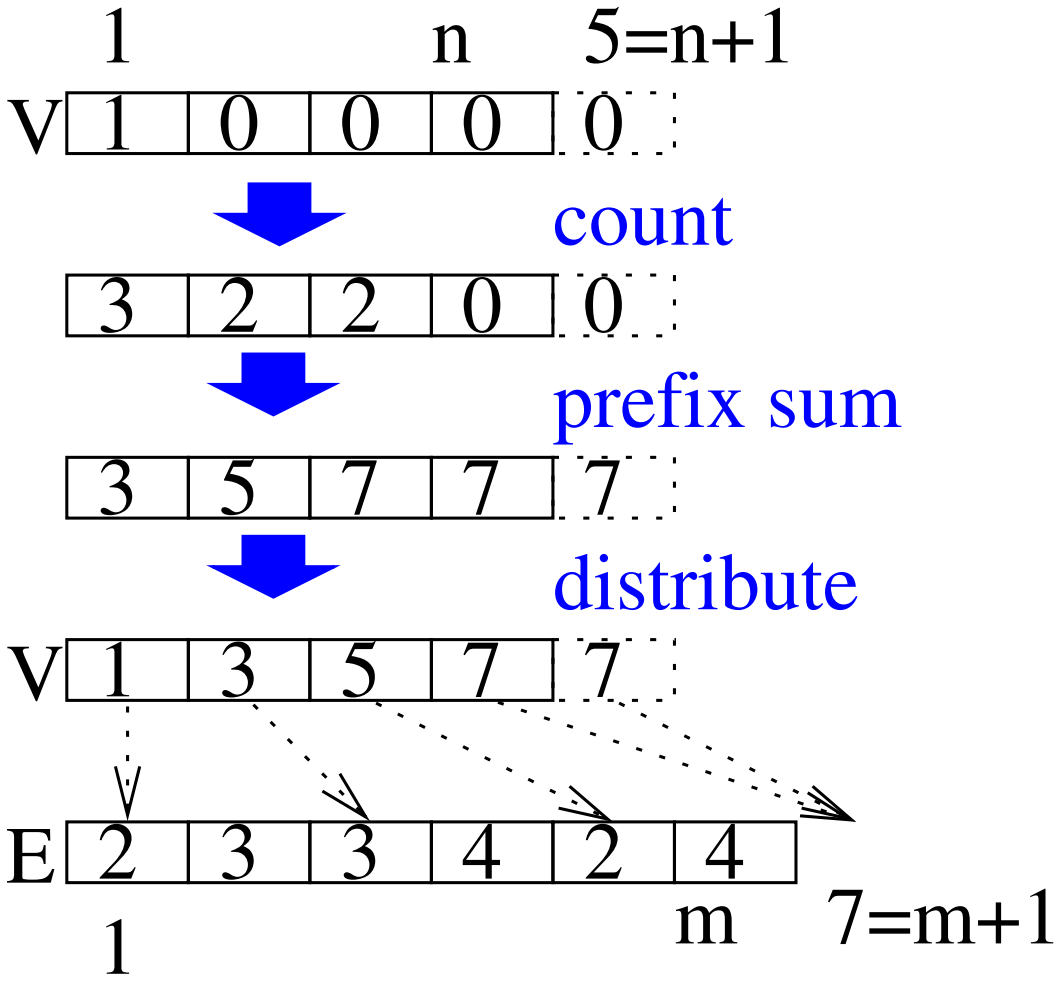
return (V,E)





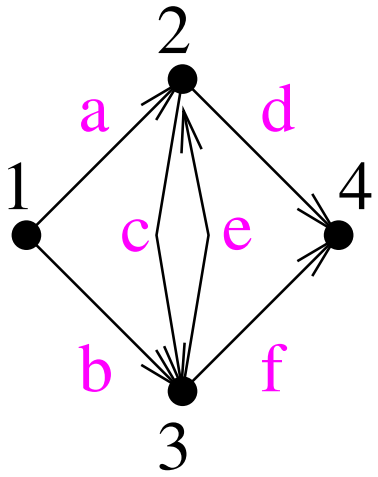
Beispiel

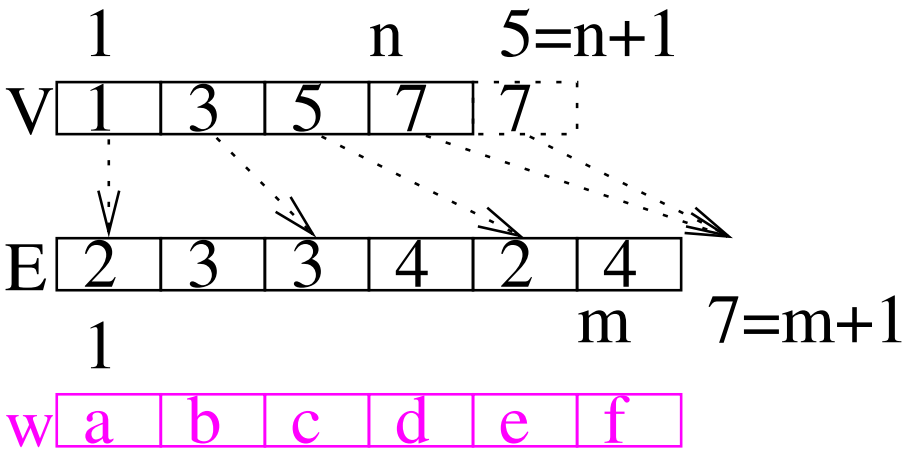

2
1
4
3


1
n
5=n+1
V
1 0 0 0 0
count
3 2 2 0 0
prefix sum
3 5 7 7 7
distribute
V
1 3 5 7 7
E
2 3 3 4 2 4
1
m
7=m+1

Operationen für Adjanzenzfelder

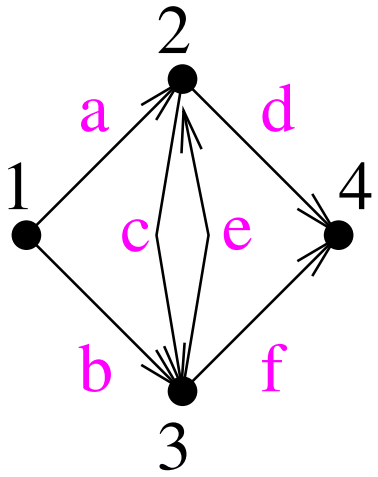
- Navigation: einfach
- Kantengewichte: E wird Feld von Records (oder mehrere Felder)
- Knoteninfos: V wird Feld von Records (oder mehrere Felder)

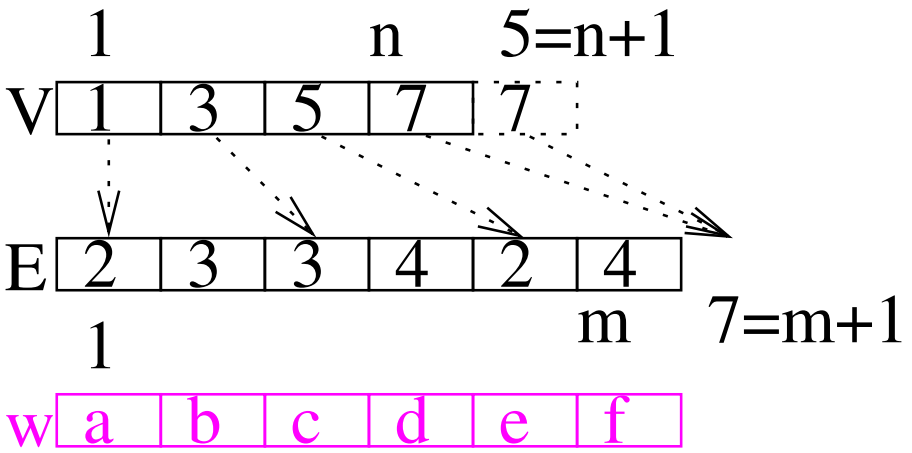




Operationen für Adjanzenzfelder

- Navigation: einfach
- Kantengewichte: E wird Feld von Records (oder mehrere Felder)
- Knoteninfos: V wird Feld von Records (oder mehrere Felder)
- Eingehende Kanten: umgedrehten Graphen speichern
- Kanten löschen: explizite Endindizes
- Batched Updates:
  neu aufbauen





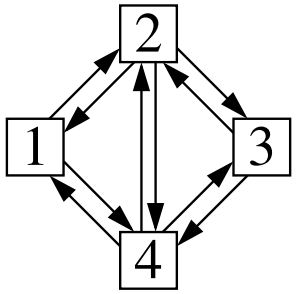
Kantenanfragen

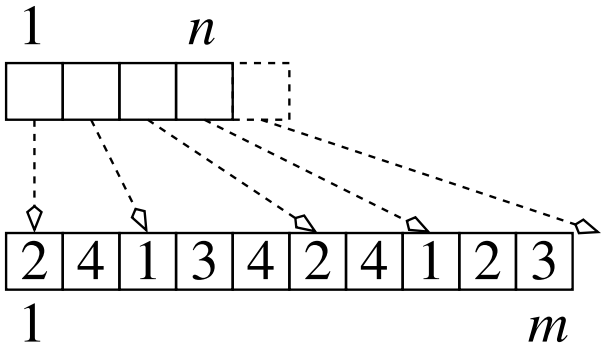
Hashtabelle H_E speichert (ggf. zusätzlich) alle Kanten.
Unabhängig von der sonstigen Graphrepräsentation

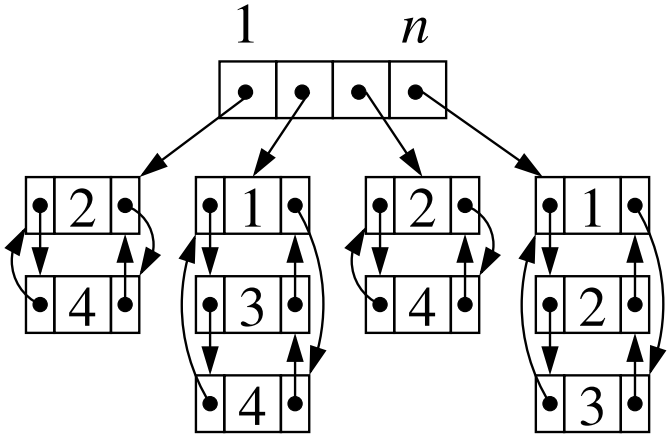
Adjazenzlisten

speichere (doppelt) verkettete Liste adjazenter Kanten für jeden Knoten.

- \+ einfaches Einfügen von Kanten
- \+ einfaches Löschen von Kanten (ordnungserhaltend)
- – mehr Platz (bis zu Faktor 3) als Adjazenzfelder
- – mehr Cache-Misses







Adjazenzlisten aufrüsten

- Knotenlisten für Knotenupdates
- Eingehende Kanten
- Kantenobjekte (in globaler Kantenliste)
- Zeiger auf Umkehrkante