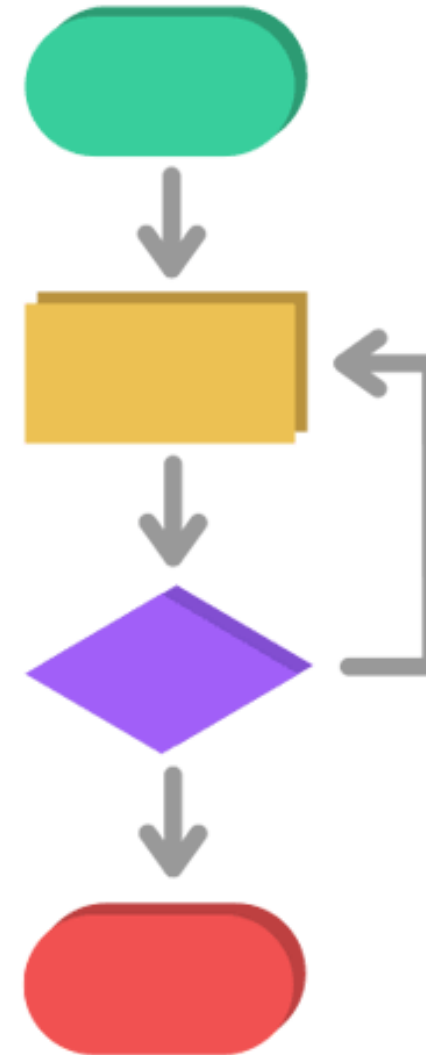


# Algorithmen Tutorium 7

Beginn: 16:15



# Organisation

---

# Bellman-Ford

---

- Dijkstra: Einen/Alle kürzesten Pfade **VOM** Start
- **Alle** kürzesten Pfade?
- Beobachtung:
  - Kürzeste Pfade haben maximal Länge  $|V|$  (ohne negative Zyklen)
  - Bilden von Pfaden der Länge  $n$  einfach (z.B. Adjazenzmatrix-Multiplikation)

# Bellman-Ford

---

- Setze Distanz für alle Knoten auf unendlich
- Wiederhole maximal  $n-1$  mal
  - Für alle Kanten  $(u,v)$ 
    - wenn  $(\text{Distanz}(u) + \text{Gewicht}(u, v)) < \text{Distanz}(v)$ 
      - $\text{Distanz}(v) = \text{Distanz}(u) + \text{Gewicht}(u, v)$

# Bellman-Ford

---

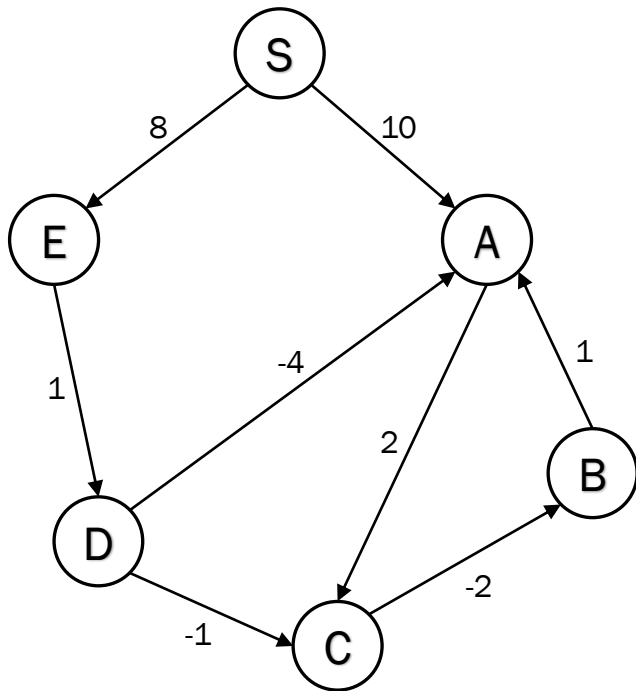
- Bonus: Negative Zyklen erkennen
  - Negative Zyklen werden um jeden beteiligten Knoten gebildet
  - Erweitere die Pfade ein weiteres Mal
  - Bildet Pfade der Länge  $|V| + 1 \Rightarrow$  Muss Zyklen enthalten
  - Kosten verringern sich  $\Rightarrow$  **negativer Zyklus!**

$\Rightarrow$  Erkennbar, ob BF korrekt ausgeführt wurde

- Backtracking:
  - Speichere zusätzlich zu den Kosten den Vorgänger

# Bellman-Ford auf Papier

---

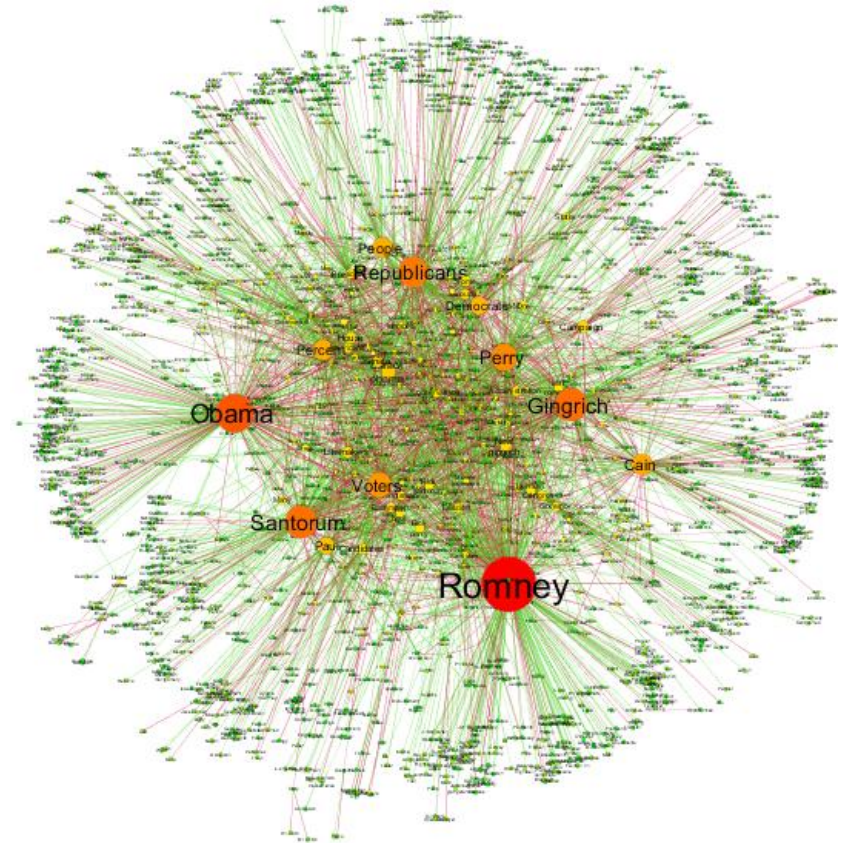


Knoten	Distanz	Vorgänger
S		
A		
B		
C		
D		
E		

# Graphenmetriken

---

- Dienen zur Abschätzung von Eigenschaften
  - Wie zentral (topologisch)?
  - Wie gut erreichbar?
  - Wie „wichtig“?
  - Welche Cluster?



Thinkbig-project, CC BY 4.0 <<https://creativecommons.org/licenses/by/4.0/>>, via Wikimedia Commons

# Misc I

---

- Pseudocode sollte klar und genügend ausführlich sein!
  - Definiert/beschreibt Variablen und Funktionen
  - Kurze Prosa/Kommentare:
    - „Der Algorithmus ist quasi Bellman-Ford, nur ...“
- Laufzeiten
  - Immer anhand EURES Codes/Beschreibung
  - Bei Referenz auf Algorithmen (ohne eigene Implementierung): Laufzeiten aus der VL
    - „Wir nutzen Bellman-Ford aus der VL mit einer Laufzeit von  $O(nm)$ “
  - Auf alle Änderungen eingehen
  - Nur weil es ein Einzeiler ist, ist es nicht in  $O(1)$



# Misc II (Programmieraufgaben)

---

- Ausführbar entweder mit:
  - cmake (**Makefile vorhanden**)
  - gcc (**Befehl zum ausführen als Kommentar in der main.cpp**)!
  - replit.com (**Link zum Projekt**)
- Korrekt ausführbar sein (Keine Endlosschleifen, Exceptions, ...)
- Wenn es nicht funktioniert => Auskommentieren & Kommentar (Gibt evtl. Teilpunkte)
- **Nicht kompilierbare oder korrekt ausführbare Programme geben sofort 0P**

