

Wahlalgorithmen

Motivation

Grundlagen

Chang-Roberts-Algorithmus

Wellenverfahren

Adoptionsverfahren



Motivation

Systemmodell

- Verteiltes System mit potenziell sehr vielen Knoten

Knoten

- Gesamtzahl aller Knoten ist unbekannt
- Jeder Knoten hat eindeutige ID
- Nicht alle Knoten sind kontinuierlich Teil des Systems

Unterschiedliche Netztopologien

- Ring
- Baum
- Beliebige Strukturierung

→ Nicht jeder Knoten ist notwendigerweise mit jedem anderen verbunden

Herausforderungen

- Wie kann eine Wahl in nicht vollvermaschten Systemen realisiert werden?
- Wie lässt sich die Effizienz durch Wissen über die Netztopologie steigern?
- Wie wählt man einen Anführer in Systemen mit komplexer Netztopologie?



Motivation

Problem: Wahl eines Anführerknotens

Beispielszenarien

- Koordinierung verteilter Aktionen
- Erzeugung systemweit eindeutiger Token

Anlässe

- Initialisierung im Rahmen des Systemstarts
- Neukonfigurierung als Reaktion auf Fehler

Verwandtes Problem: Bestimmung einer maximalen Knoten-ID

Anforderungen

- *Eindeutigkeit*: Zu jedem Zeitpunkt ist maximal ein Knoten der Anführer
- *Terminierung*: Bestimmung des Anführers erfolgt in endlicher Zeit

Zusätzliche Kriterien (Beispiele)

- Deterministischer Wahlalgorithmus
- Benachrichtigung aller Knoten über das Ende bzw. Ergebnis der Wahl



Sequentielle Verfahren

Tiefensuche auf Bäumen

```
public class VSNode {  
    private final int id = [ID des lokalen Knotens];  
    private final List<VSNode> children = [Liste der Kindknoten];  
  
    public int findMaxID() {  
        int maxID = id;  
        for(VSNode child: children) {  
            int childMaxID = child.findMaxID();  
            if(maxID < childMaxID) maxID = childMaxID;  
        }  
        return maxID;  
    } }
```

Tiefensuche auf beliebigen Topologien (Beispiel)

- Ein Initiatorknoten „empfängt“ ein Token über eine virtuelle Kante k_{init}
- Empfang des Tokens durch Knoten i über eine Kante k
 - Erstmaliger Empfang: Eintrag von i ins Token, merken der Kante $k_{first} := k$
 - Weitergabe des Tokens an einen Nachbarn, der das Token noch nicht hatte
 - Falls kein solcher Nachbar existiert: Senden des Tokens über Kante k_{first}
- Terminierung, sobald Initiator das Token über die Kante k_{init} „sendet“



Vorbereitung

- Ermittlung einiger Adressen anderer Knoten für Verbindungsaufbau
- Beispiel: Nutzung einer Registry

Verbindungsaufbau

- Verwaltung eines lokalen Levels
- Wurzelknoten des Baums hat Level 0
- Elternknoten: Nachbar mit kürzester Distanz zum Wurzelknoten

Wiederholung der Auswahl bei Abbruch der Elternverbindung

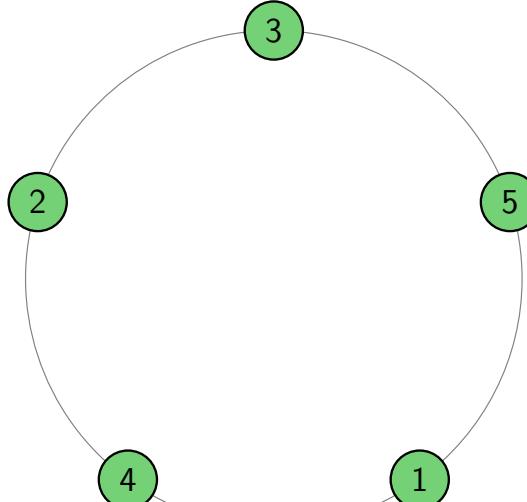
```
private int level = -1;
private VSNode parent = null;

public void connect(List<VSNode> nodes) {
    for(VSNode node: nodes) {
        // Verbindungsaufbau
        boolean connected = [Aufbau der Verbindung];
        if(!connected) continue;

        // Bestimmung des Elternknotens
        if((level < 0) || (level > node.level)) {
            level = node.level + 1;
            parent = node;
        }
    }
}
```



Beispiel



Voraussetzungen

- Anordnung aller Knoten in einem (logischen) unidirektionalen Ring
- Definition einer totalen Ordnung auf Knoten-IDs (z. B. $ID \in [1, \dots, n]$)

Bestimmung des aktiven Knotens mit der höchsten ID

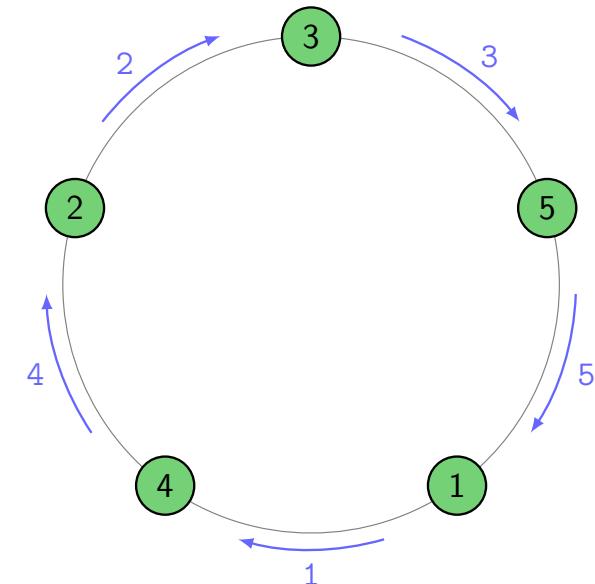
- Start des Wahlalgorithmus
 - Annahme: Alle Knoten starten Algorithmus zur selben Zeit
 - Jeder Knoten sendet eigene Knoten-ID im Ring weiter
 - Jeder Knoten wird wählbar
- Empfang einer ID i
 - Falls $i >$ eigene ID Knoten sendet i weiter und ist nicht mehr wählbar
 - Falls $i <$ eigene ID Nachricht ignorieren
 - Falls $i =$ eigene ID Falls Knoten wählbar, dann ist er als Anführer gewählt
Sonst: Nachricht ignorieren

Literatur

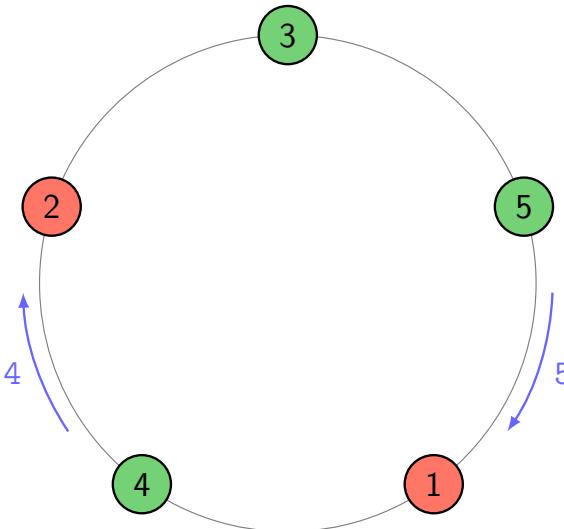
Ernest Chang, Rosemary Roberts
An improved algorithm for decentralized extrema-finding in circular configurations of processes
Communications of the ACM, 22(5):281–283, 1979.



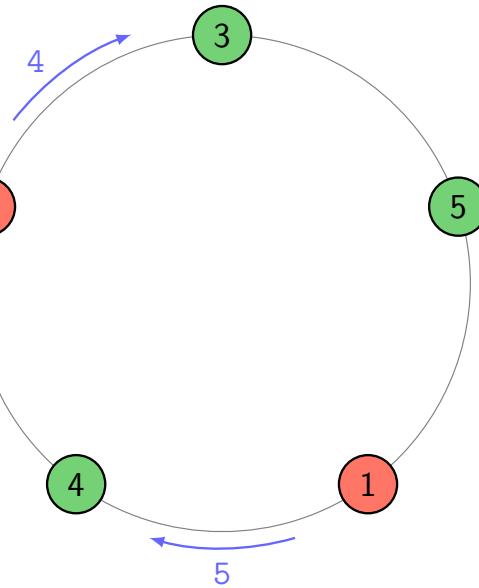
Beispiel



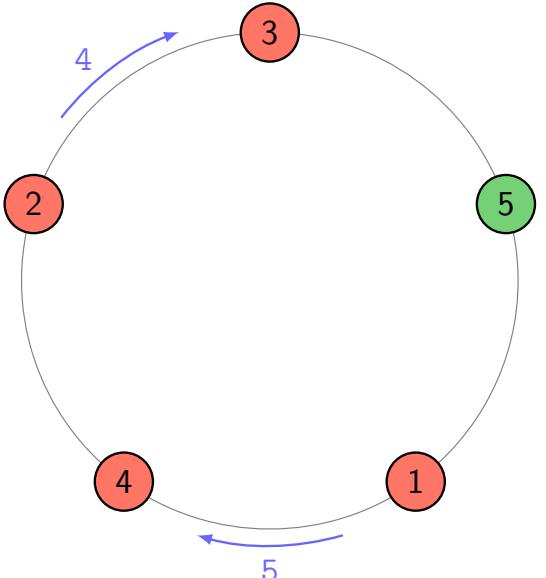
Beispiel



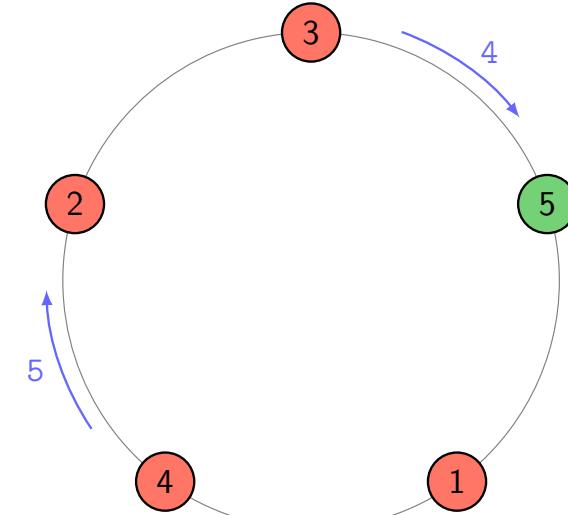
Beispiel

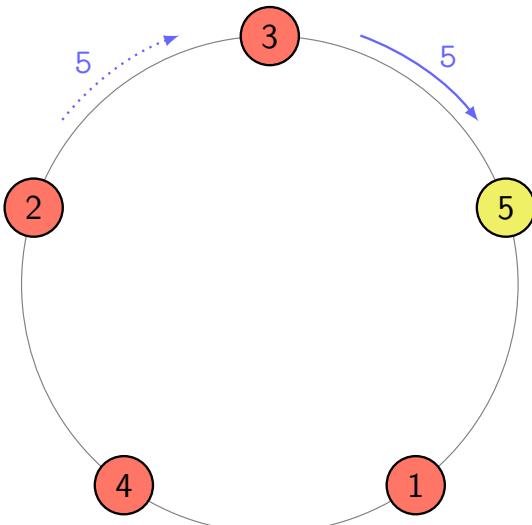


Beispiel



Beispiel





Komplexität

- Annahmen bei der Analyse der Zeitkomplexität
 - Bearbeitungsdauer von Nachrichten ist vernachlässigbar
 - Nachrichten kommen innerhalb einer maximalen Nachrichtenlaufzeit an
- Bester Fall: Knoten sind den IDs nach aufsteigend angeordnet
 - n Nachrichtenlaufzeiten
 - $n + (n - 1) \cdot 1 = 2n - 1$ Nachrichten $\rightarrow O(n)$
- Schlimmster Fall: Knoten sind den IDs nach absteigend angeordnet
 - n Nachrichtenlaufzeiten
 - $\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}$ Nachrichten $\rightarrow O(n^2)$
- Durchschnittlicher Fall
 - Annahme: Alle $(n - 1)!$ möglichen ID-Verteilungen gleichwahrscheinlich
 - n Nachrichtenlaufzeiten
 - Nachrichtenaufwand: $O(n \log n)$ (siehe [Chang et al.])

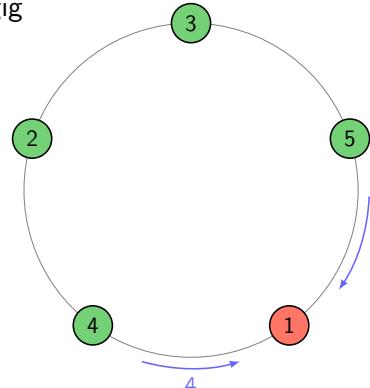
- Bemerkungen
 - Einfacher Algorithmus auf Basis eines einfachen Systemmodells
 - Teilnehmer müssen die Gesamtzahl aller Knoten n nicht kennen
- Mögliche Erweiterung
 - Anführer sendet nach seiner Wahl eine Benachrichtigung durch den Ring
 - Sauberes Beenden des Wahlalgoritmus auf allen Knoten möglich
- Korrektheit
 - Eindeutigkeit
 - Nachricht mit einer bestimmten ID wird von genau einem Knoten erzeugt
 - Knoten mit höchster ID leitet nur seine eigene Nachricht weiter
 - Ein Knoten wird nur gewählt, wenn seine ID von allen weitergereicht wurde
 - Terminierung
 - Die Nachricht mit der größten ID wird von jedem Knoten weitergereicht
 - Nach n Schritten kommt die größte ID wieder beim Ursprungsknoten an
 - Achtung: Terminierung ist bei Ausfällen von Knoten nicht sichergestellt

Initiierung

- Veränderte Startbedingung
 - Gleichzeitiger Start des Algorithmus auf allen Knoten unrealistisch
 - Ein Knoten startet den Algorithmus durch Senden seiner ID
 - Andere Knoten beteiligen sich erst, nachdem sie eine Nachricht erhalten
- Zeitkomplexität
 - Bester Fall
 - Knoten mit höchster ID initiiert Wahl
 - n Nachrichtenlaufzeiten
 - Schlechtester Fall
 - Knoten mit höchster ID wacht als Letzter auf
 - $(n - 1) + n = 2n - 1$ Nachrichtenlaufzeiten
- Optimierung
 - Unterdrückung der eigenen Nachricht, falls Aufwecken durch größere ID
 - Reduzierung der im besten Fall erforderlichen Nachrichten auf n

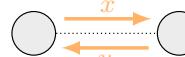
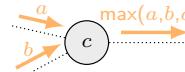
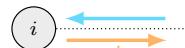
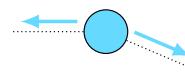
Variante

- Veränderte Architektur: Bidirektionaler Ring
- Unterschiede zur Standardvariante
 - Start: Jeder Knoten bestimmt die Senderichtung der eigenen ID zufällig
 - Jeder Knoten speichert die höchste ID i_{max} , die er bisher gesehen hat
 - Unterdrückung einer neu empfangenen ID i , falls $i < i_{max}$
 - Senderichtung der Weiterleitung abhängig von bisheriger Senderichtung einer ID
- Bemerkungen
 - Geringere Anzahl erforderlicher Nachrichten im Durchschnittsfall
 - Schlimmster Fall
 - Geringere Auftrittswahrscheinlichkeit
 - Nachrichtenaufwand weiterhin $O(n^2)$



Algorithmus

- Dedizierte Initiatorknoten senden *Explorernachrichten* an alle ihre Nachbarknoten
- Blattknoten senden bei Empfang einer Explorernachricht die eigene ID als Echonachricht zurück
- Innere Knoten mit k Kanten
 - Weiterleitung der ersten Explorernachricht in alle übrigen Richtungen
 - Nach Erhalt von Echonachrichten auf $k - 1$ Kanten: Senden einer Echonachricht e mit dem Maximum aller bisher bekannten IDs über die verbleibende Kante
 - Falls Empfang einer weiteren Echonachricht e' : Höchste ID im Netz ist Maximum der IDs von e und e'

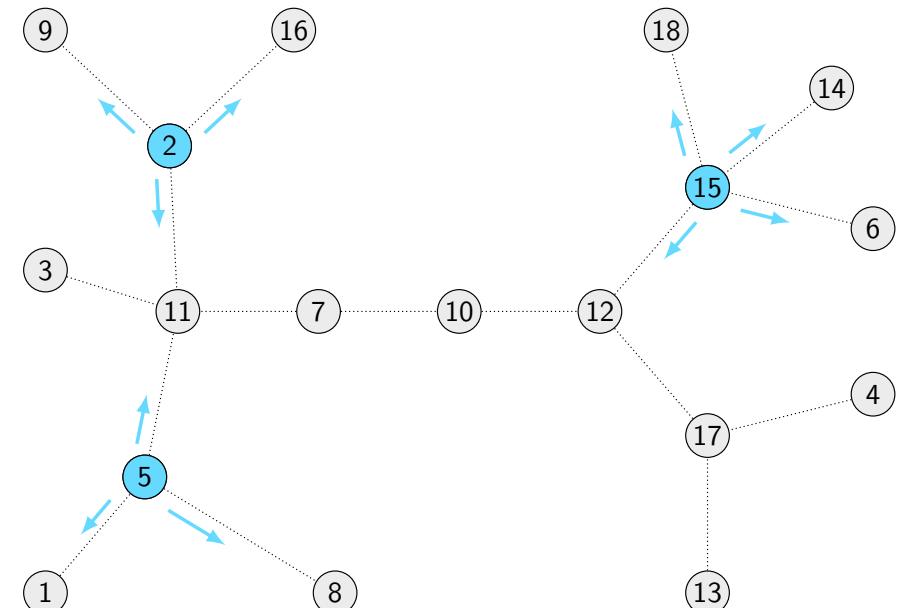


Wellenverfahren

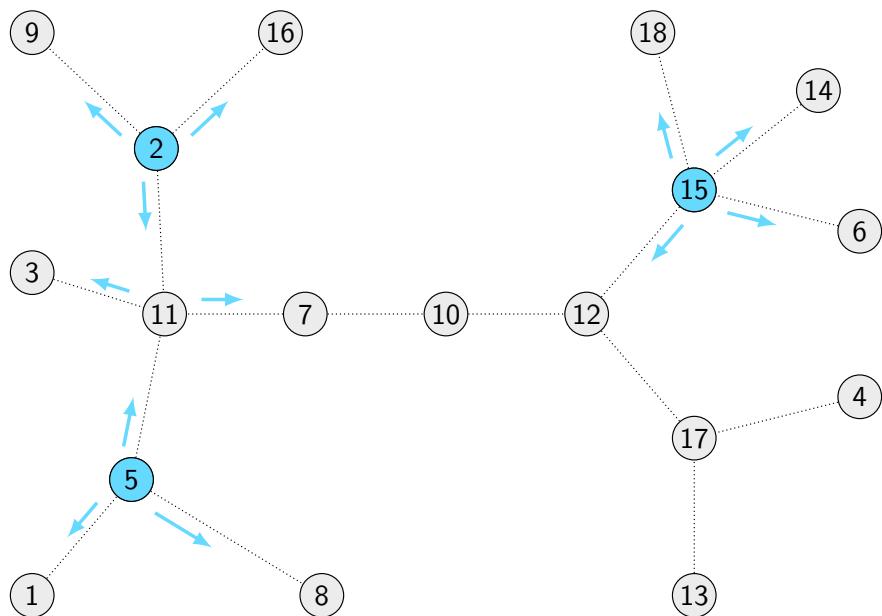
- Baumstruktur
 - Ungerichtete Kanten
 - Keine Zyklen
- Vergleich zur Wahl auf Ringen
 - Kommunikation mit unterschiedlich vielen Nachbarknoten
 - Potenzial für *Divide-and-Conquer*-Ansätze
- Wellenverfahren
 - *Explorationswelle* durchläuft den Baum bis zu den Blättern
 - *Echowelle* kommt zurück und bestimmt die höchste ID
 - *Informationswelle* teilt allen Knoten das Ergebnis mit
- Literatur
 - Friedemann Mattern
Verteilte Basisalgorithmen
Springer-Verlag, 1989.



Beispiel



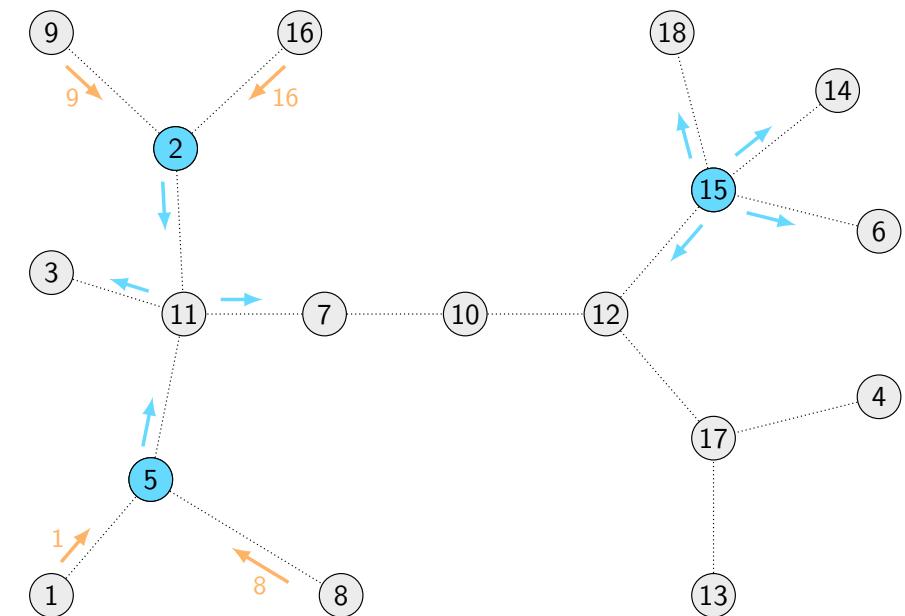
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

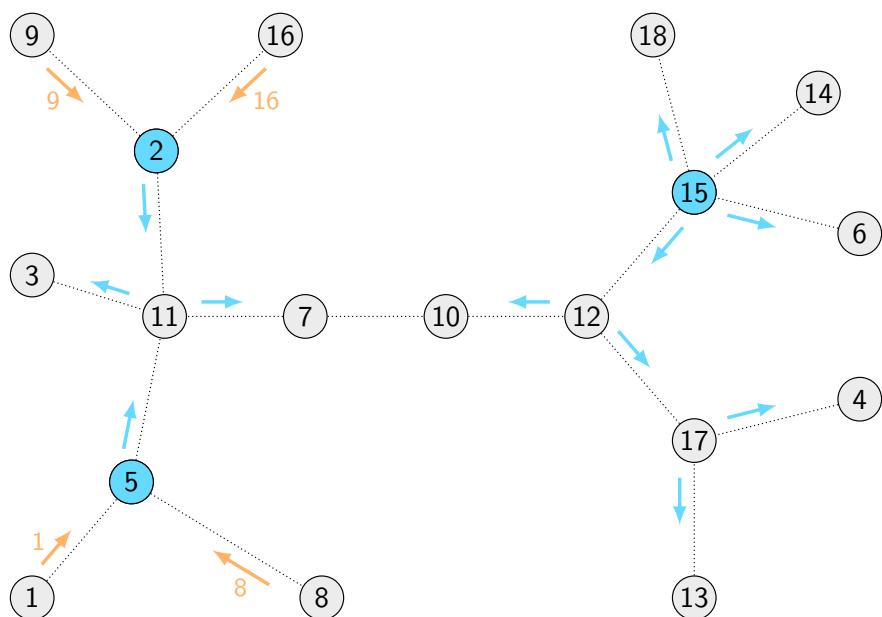
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

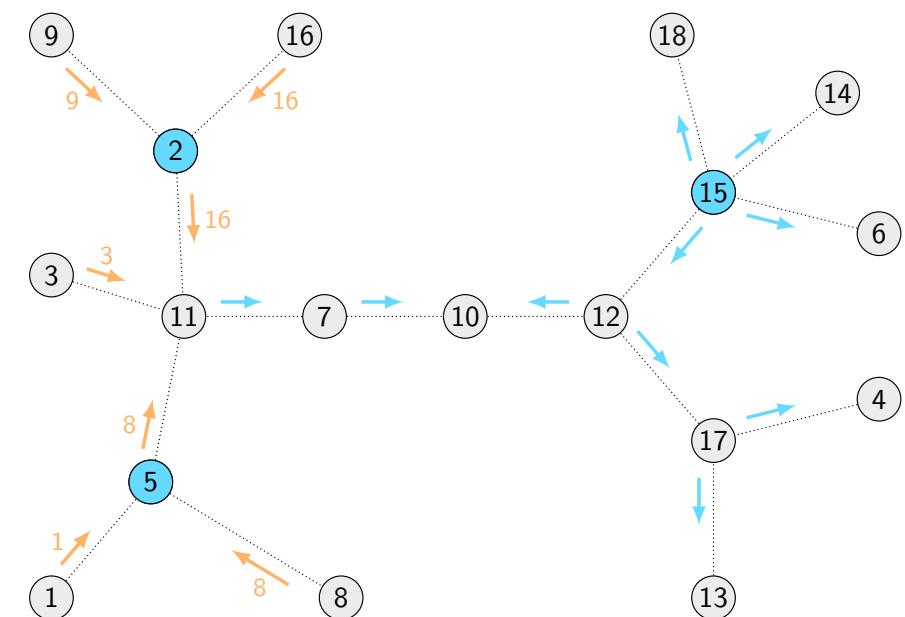
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

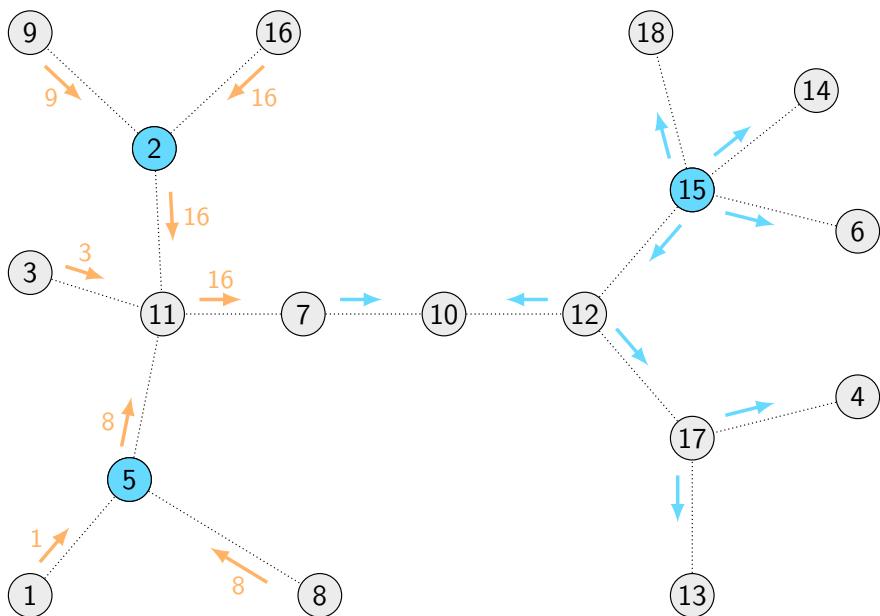
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

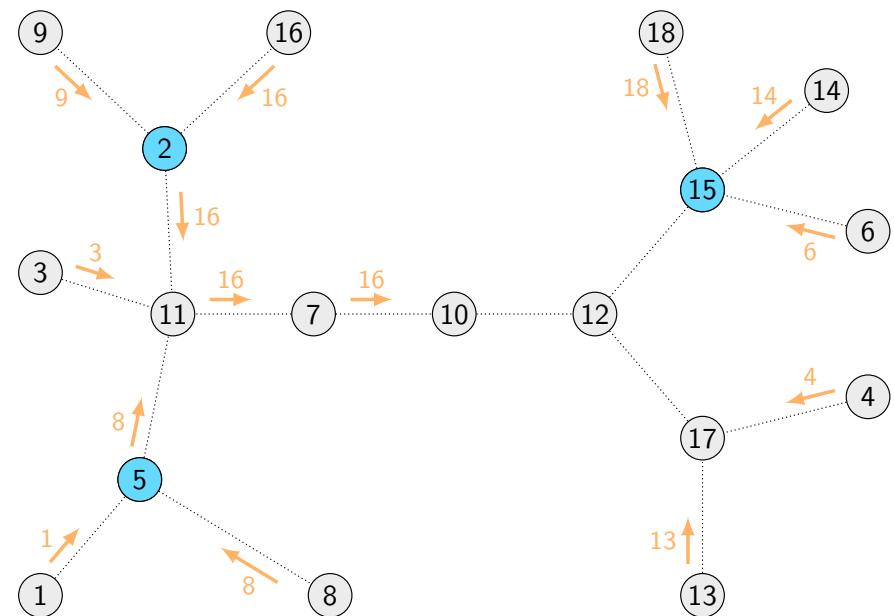
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

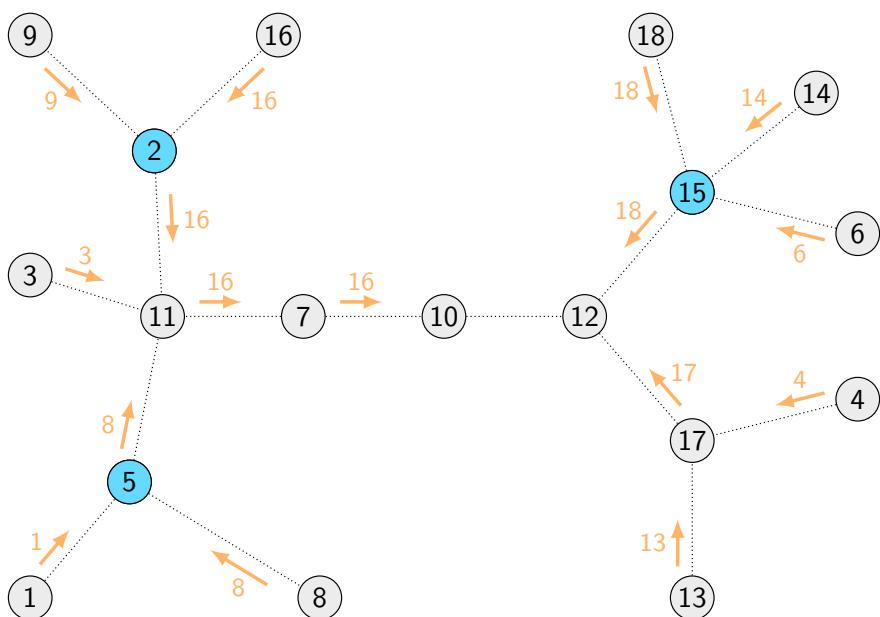
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–14

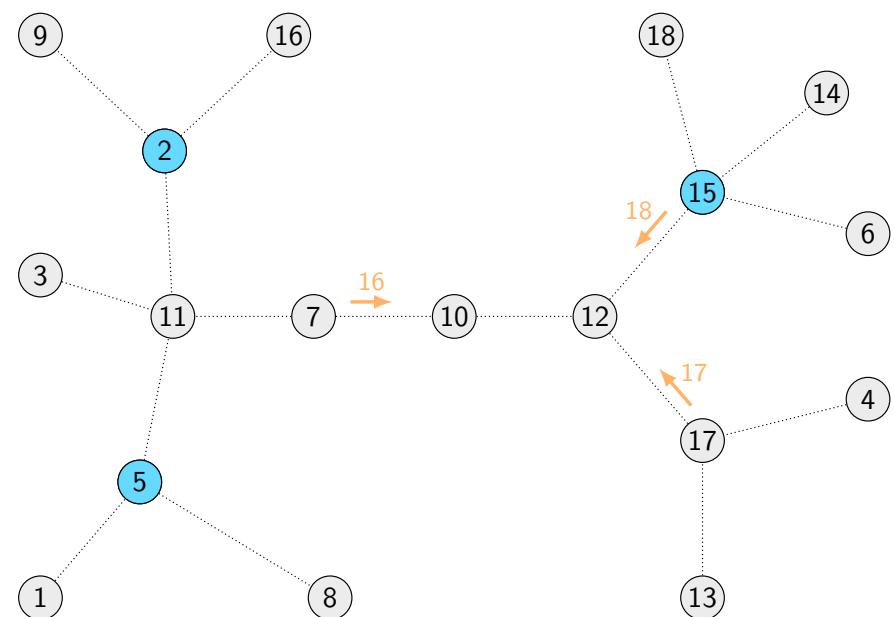
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–15

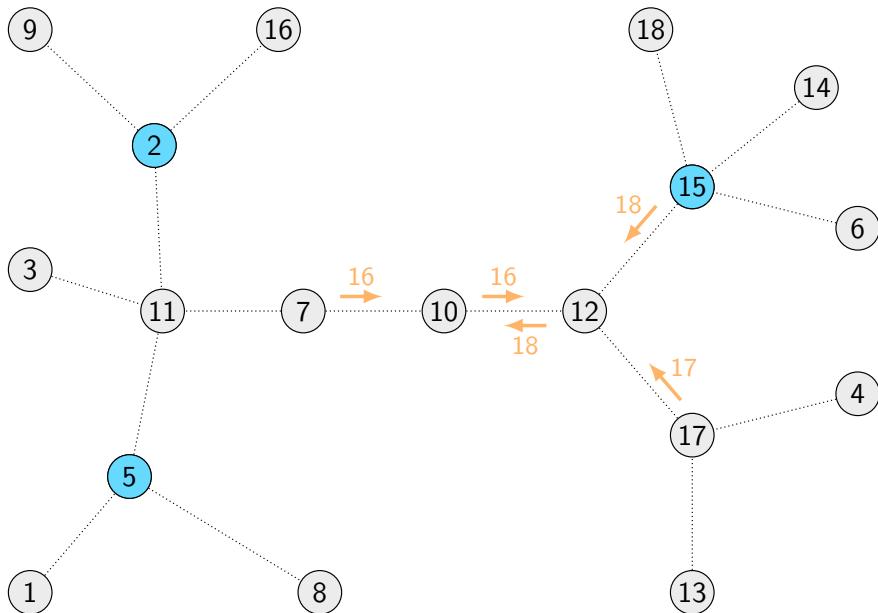
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11–15

Beispiel



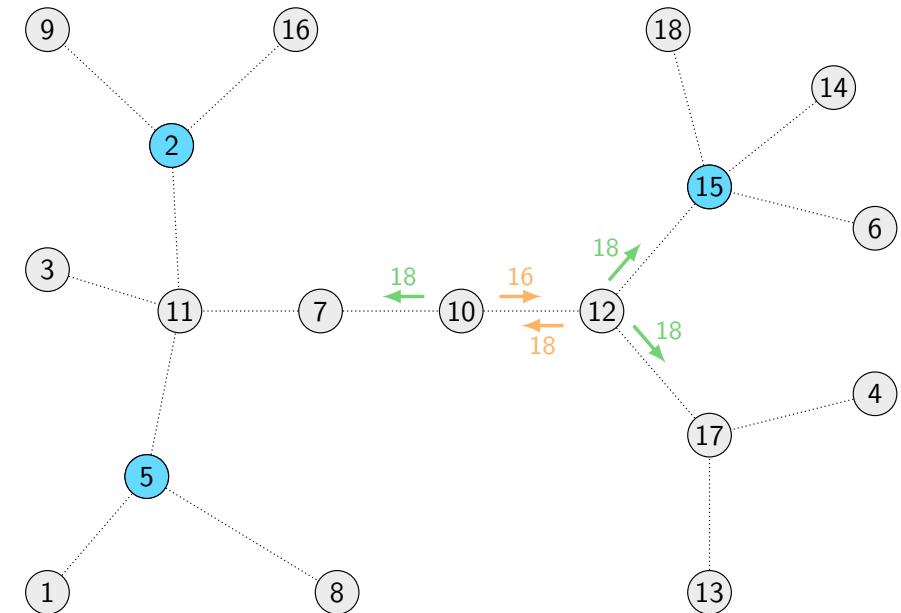
© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11-19

Adoptionsverfahren für beliebige Topologien

- Grundprinzip wie bei Wahl auf Bäumen: Explorations- und Echowelle
- Nachrichten
 - Explorernachrichten beinhalten ID des korrespondierenden Initiators
 - Echonachrichten tragen Initiator-ID der zugehörigen Explorernachricht
- Verwaltung von IDs an Knoten und Kanten
 - Speicherung der höchsten einem Knoten bekannten Initiator-ID (ID_i)
 - *Elternkante*: Kante, über die Explorernachricht mit ID; empfangen wurde
 - Markierung der Sendekante einer Explorernachricht mit Initiator-ID
- Empfang einer Explorernachricht x über Kante k mit Markierung m
 - Falls $ID_x = ID_m$ Kante k ist nicht Teil des virtuellen Baums
 - Falls $ID_x < ID_m$ Ignorieren der eintreffenden Explorernachricht x
 - Falls $ID_m < ID_x \leq ID_i$ Aktualisierung von m , Senden von x über k
 - Falls $ID_m \leq ID_i < ID_x$ *Adoption*: k wird neue Elternkante
Weiterleitung von x über bisherige Elternkante

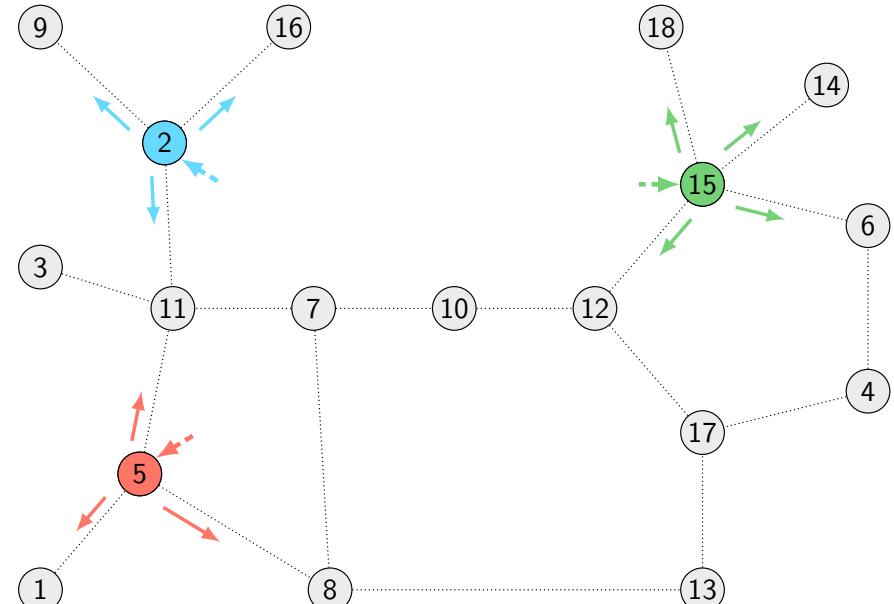
Beispiel



© td VS (SS16) Wahlalgorithmen – Wellenverfahren

11-15

Beispiel



A red circular arrow icon with a thick border, indicating a loop or a refresh function.

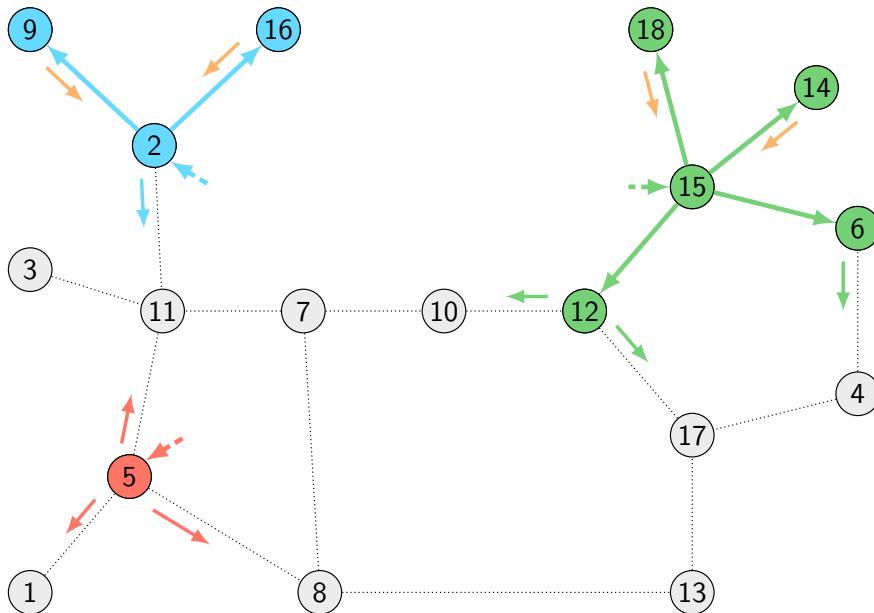
© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-19

© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-17

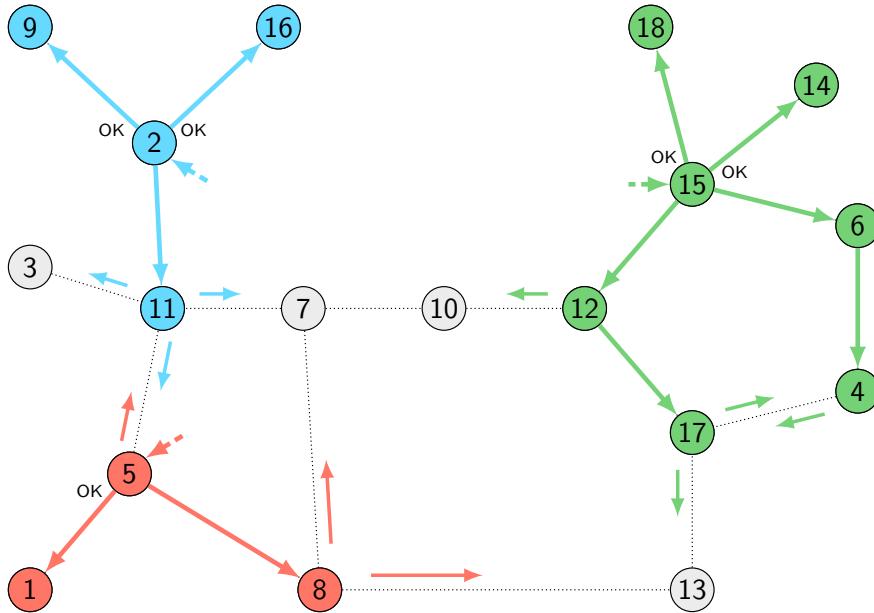
Beispiel



© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-17

Beispiel

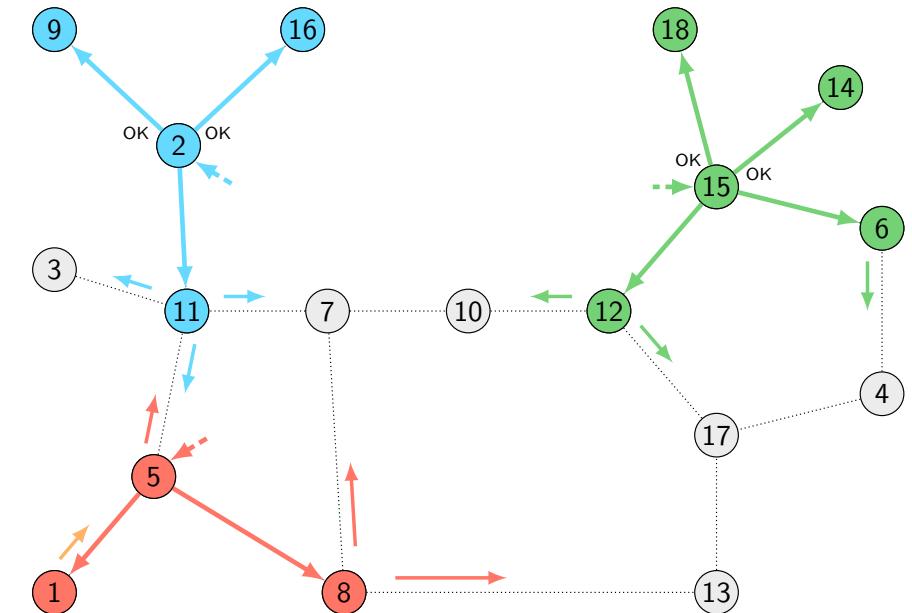


A small red icon consisting of a thick circular arrow pointing clockwise, positioned in the bottom right corner of the slide.

© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-17

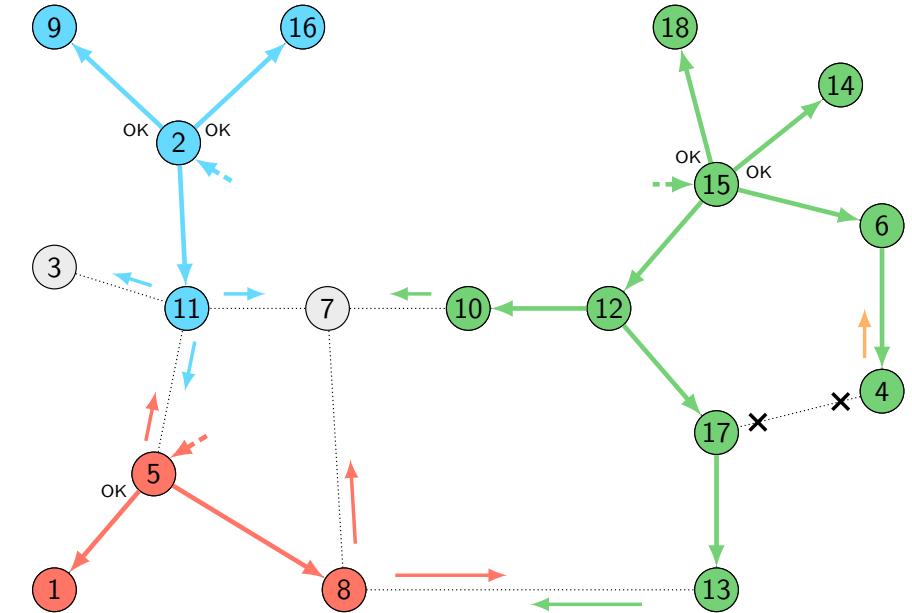
Beispiel



© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-17

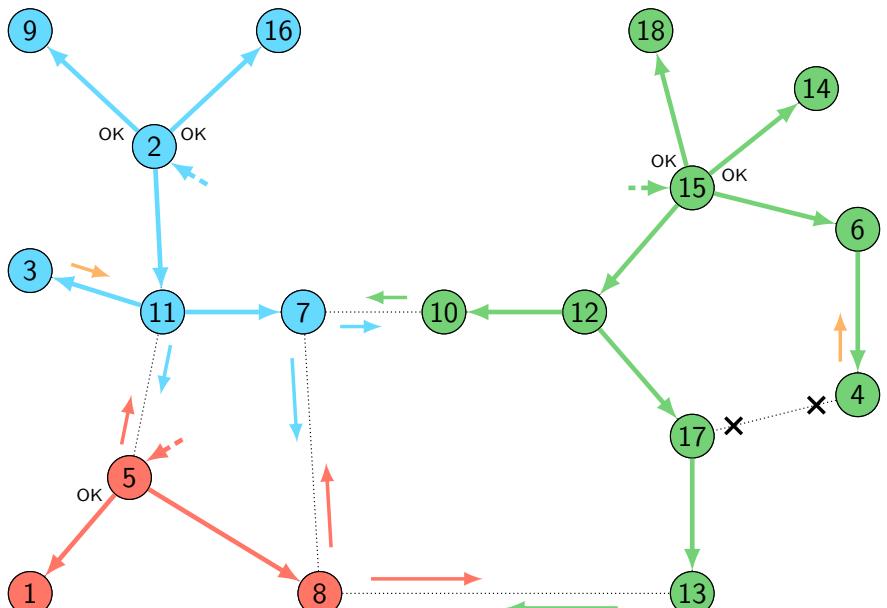
Beispiel



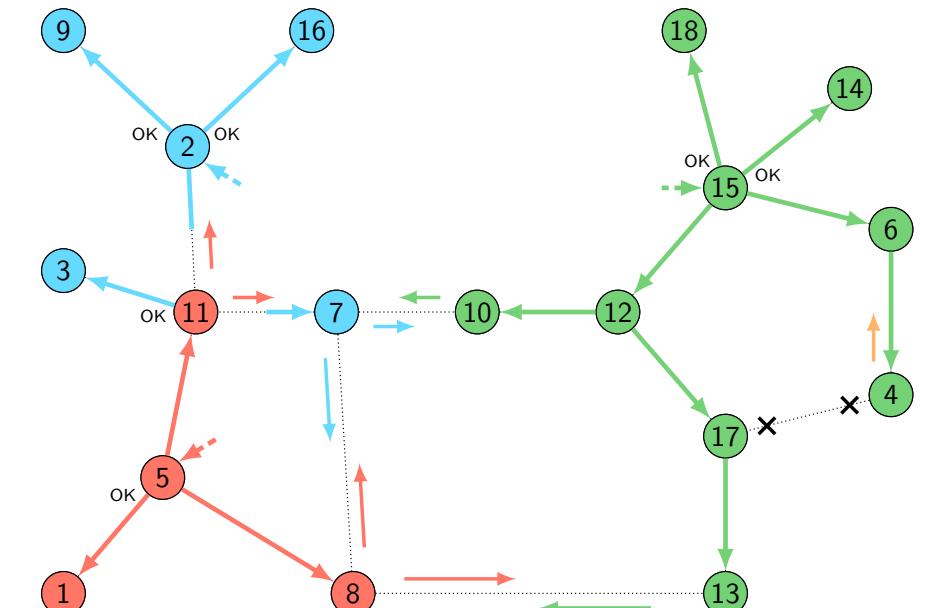
© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11-17

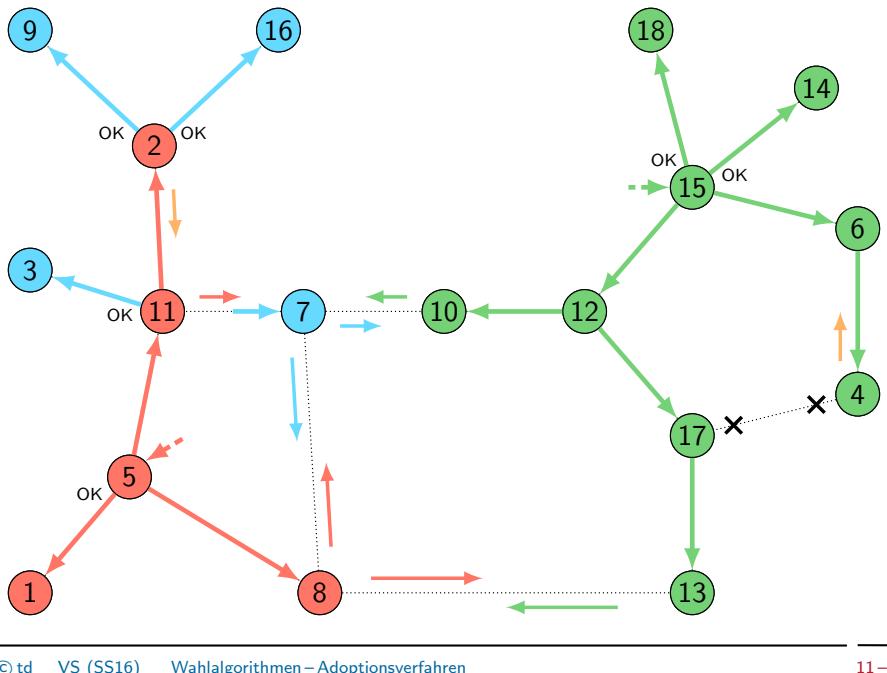
Beispiel



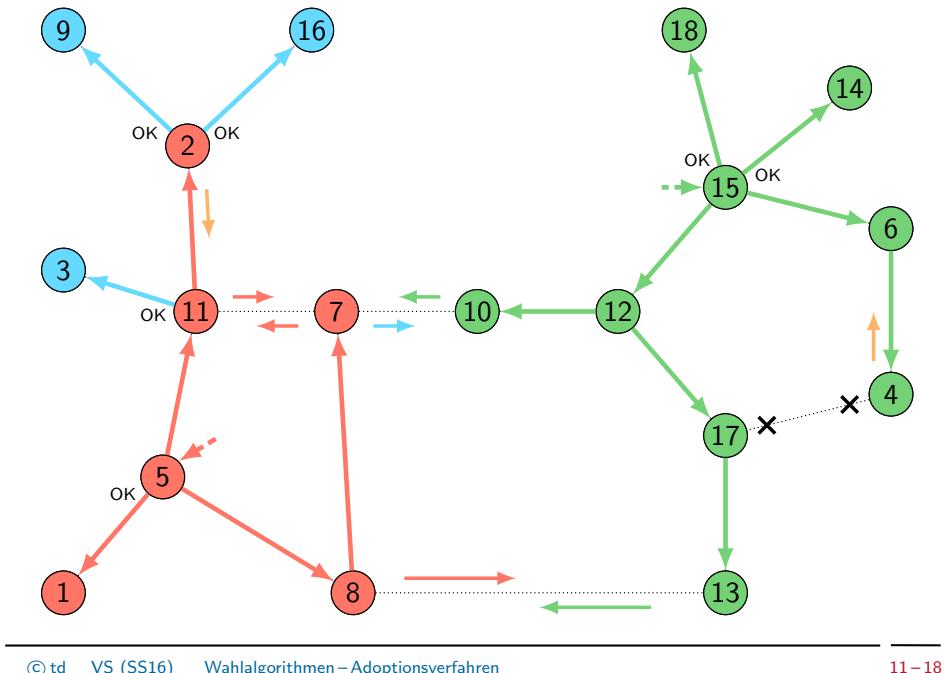
Beispiel



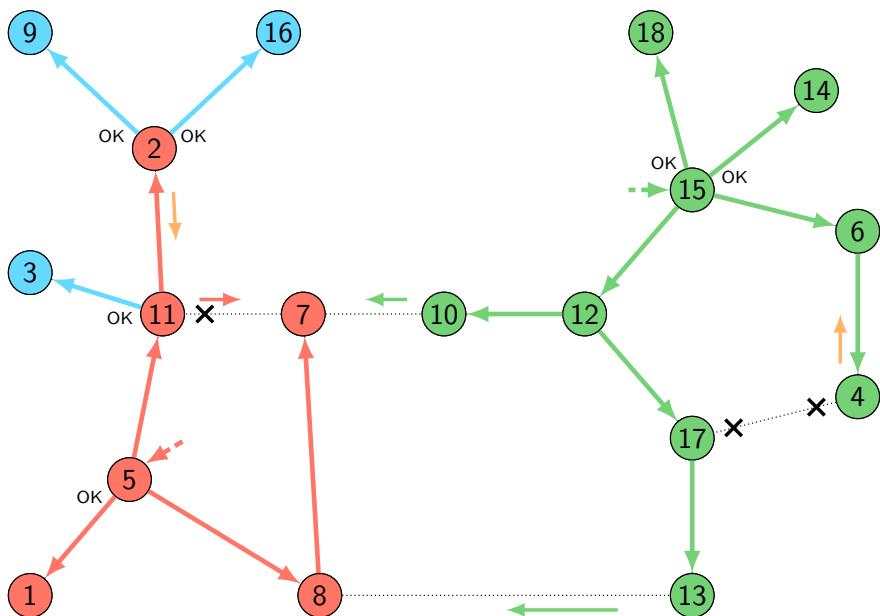
Beispiel



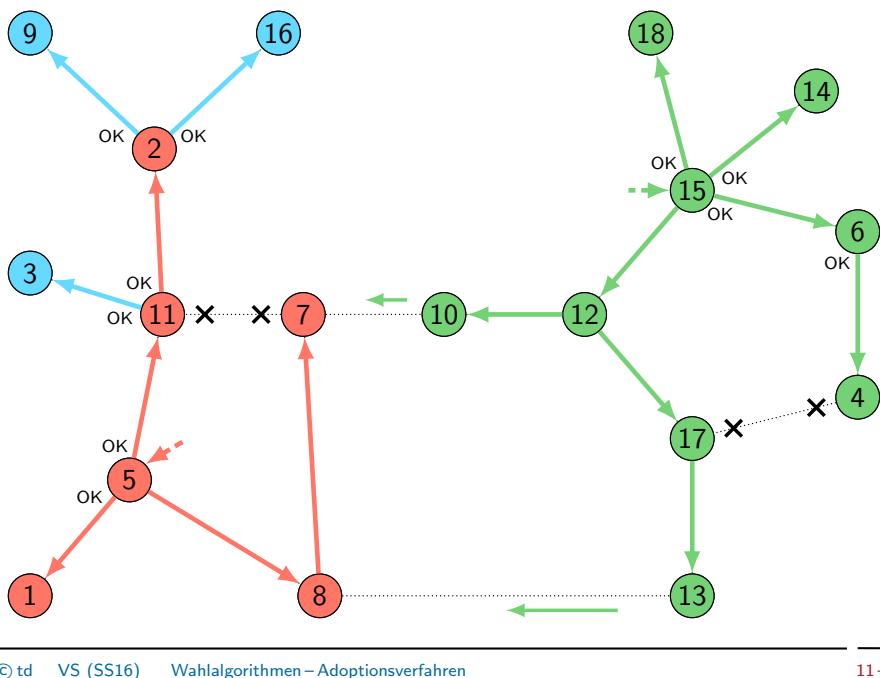
Beispiel



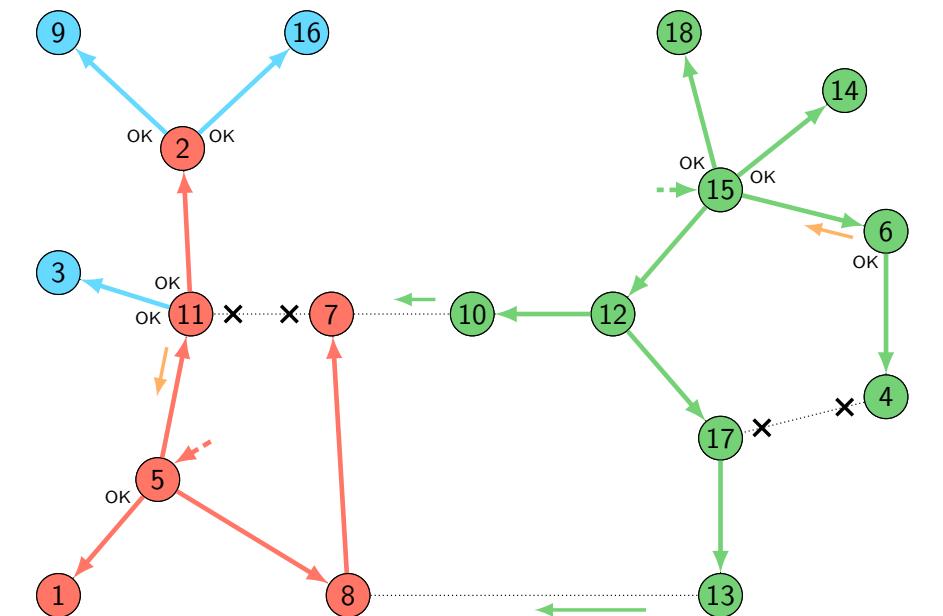
Beispiel



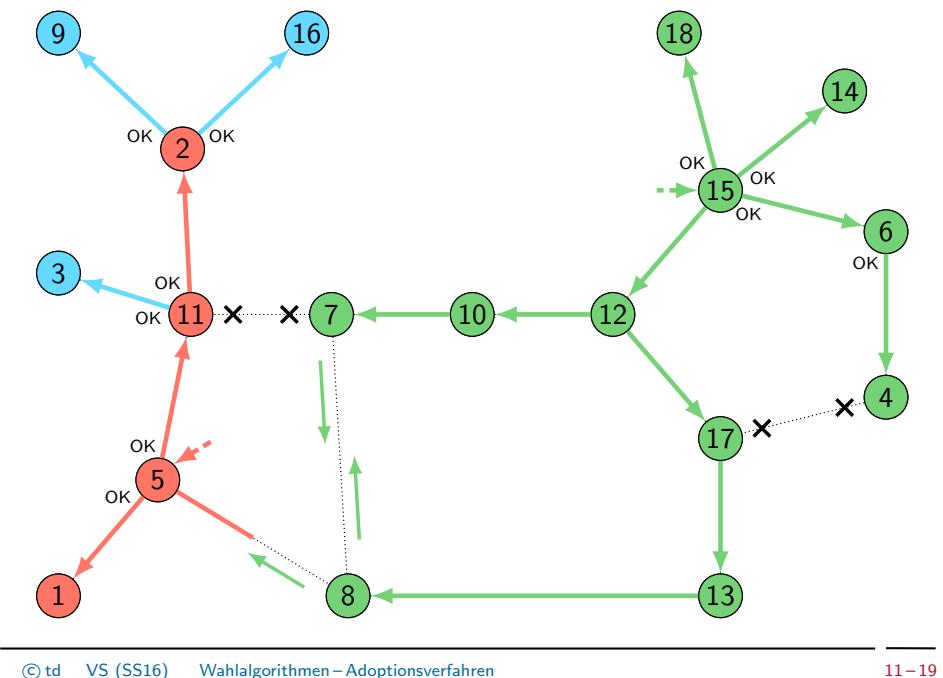
Beispiel



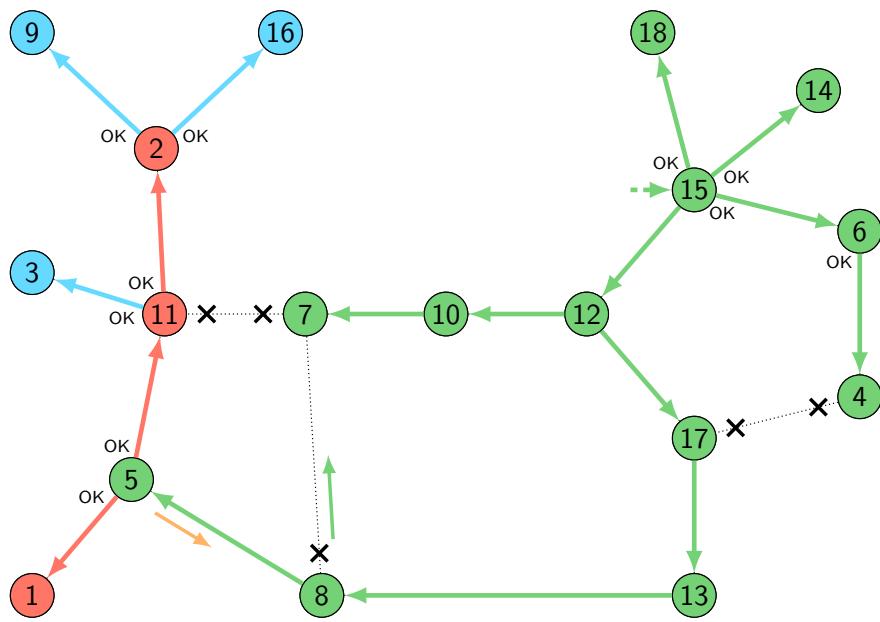
Beispiel



Beispiel



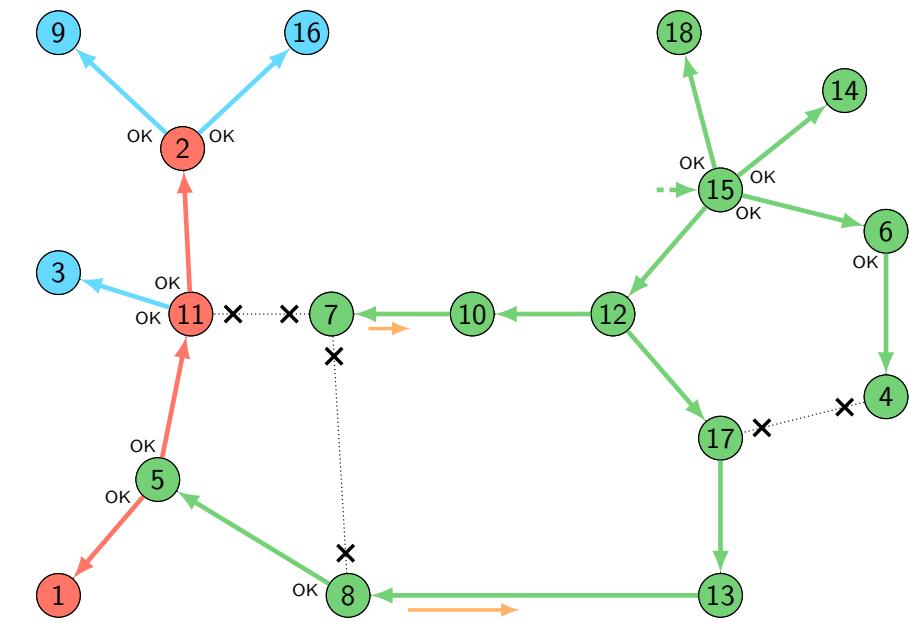
Beispiel



© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11–19

Beispiel



© td VS (SS16) Wahlalgorithmen – Adoptionsverfahren

11–19