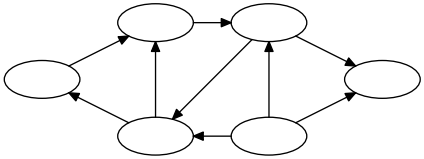
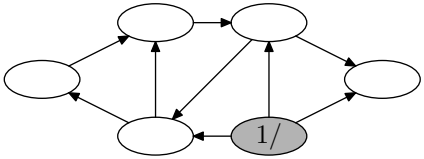
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

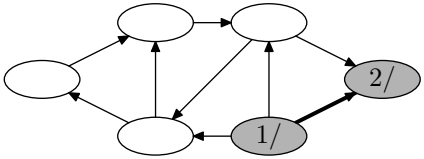
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

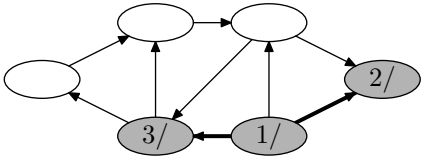
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

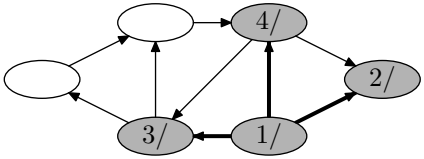
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

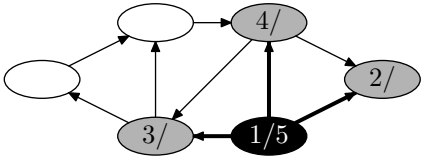
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

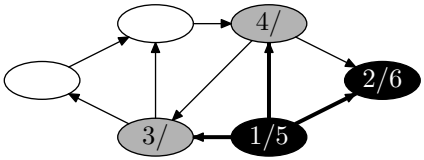
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

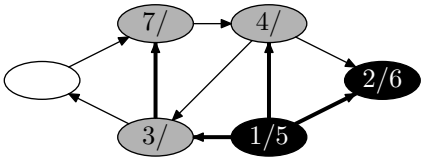
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

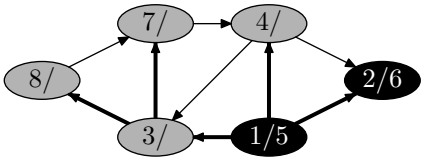
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

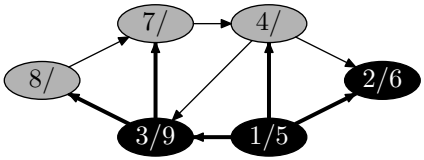
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

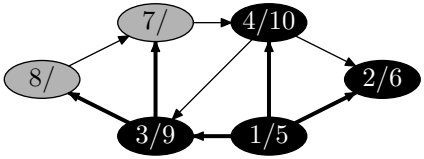
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

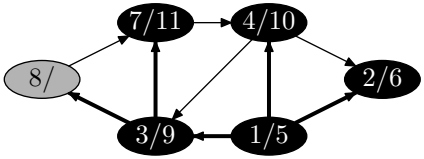
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

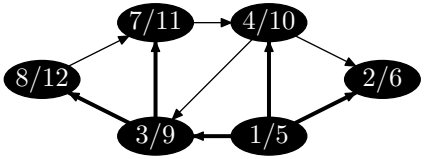
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

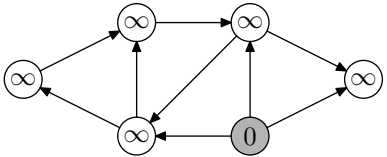
Breitensuche



Breitensuche ist das „Gegenteil“ von Tiefensuche.

- Tiefensuche: Aktive Knoten auf Stack
- Breitensuche: Aktive Knoten in FIFO-Queue
- Intelligente Suche: Weder Stack noch Queue
- Breitensuchbaum enthält kürzeste Pfade (ungewichtet)
- Discovery- und Finishzeiten wenig Anwendungen

Breitensuche



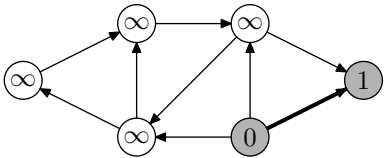
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



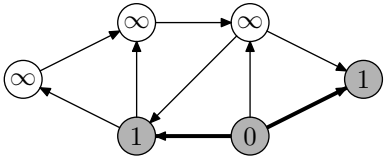
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



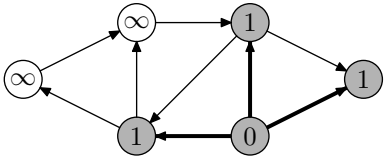
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



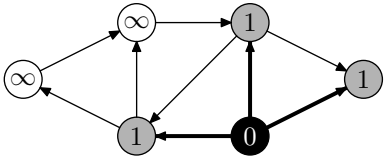
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



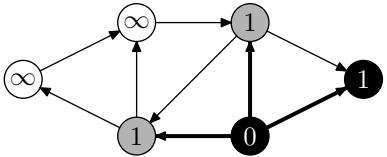
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



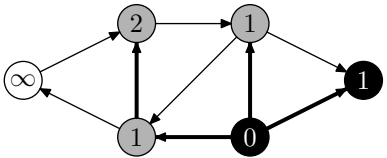
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



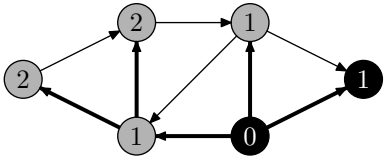
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



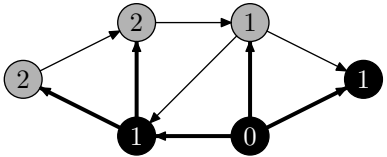
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



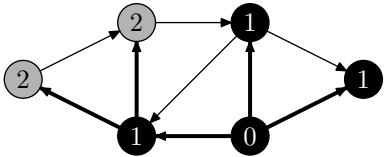
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



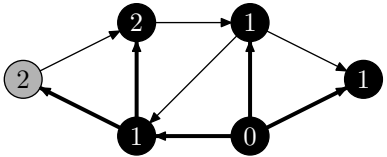
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



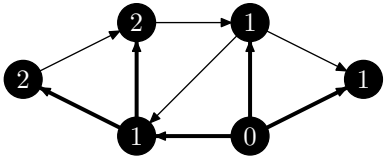
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

Breitensuche



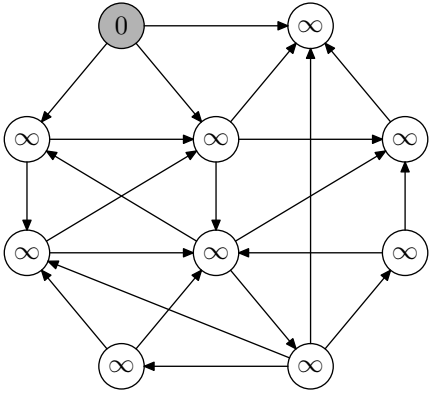
Knoten wird aktiv:

Abstand berechnen und Kante in BFS-Baum

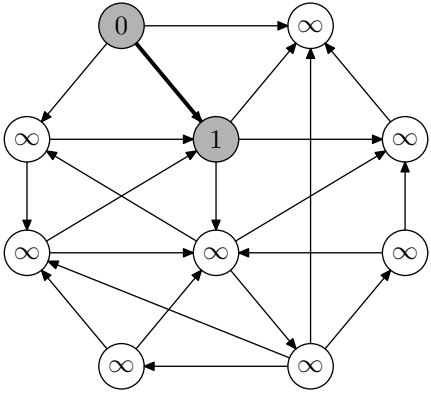
Anwendung:

Alle Abstände und kürzesten Wege von s berechnen

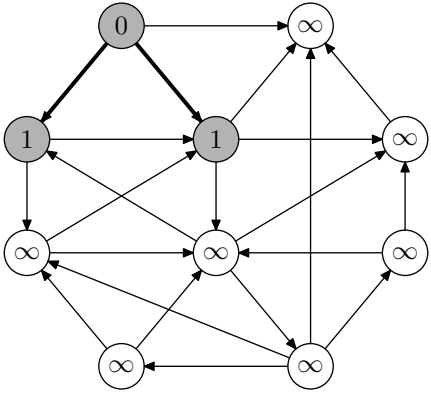
Breitensuche – Größeres Beispiel



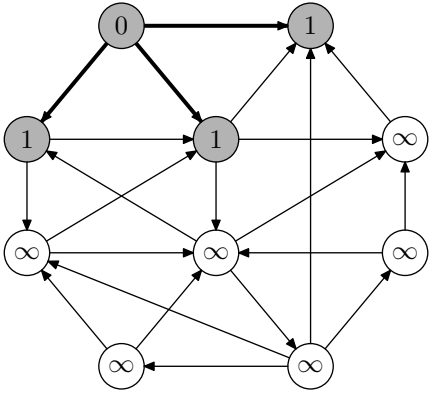
Breitensuche – Größeres Beispiel



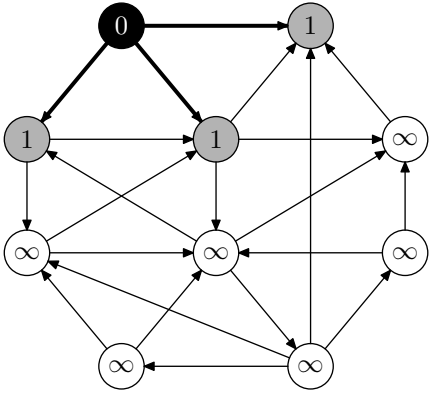
Breitensuche – Größeres Beispiel



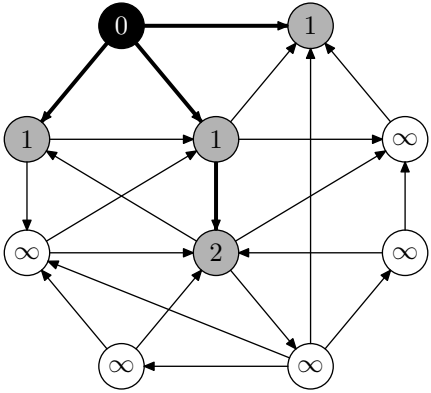
Breitensuche – Größeres Beispiel



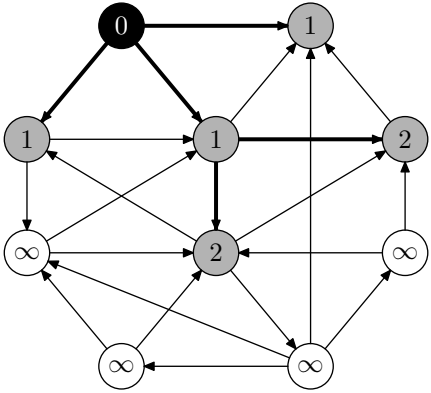
Breitensuche – Größeres Beispiel



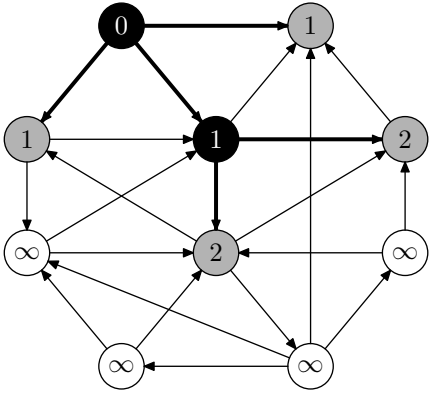
Breitensuche – Größeres Beispiel



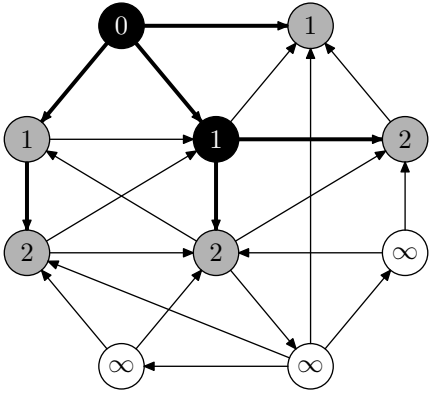
Breitensuche – Größeres Beispiel



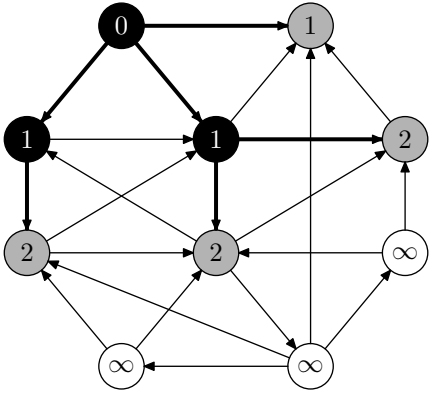
Breitensuche – Größeres Beispiel



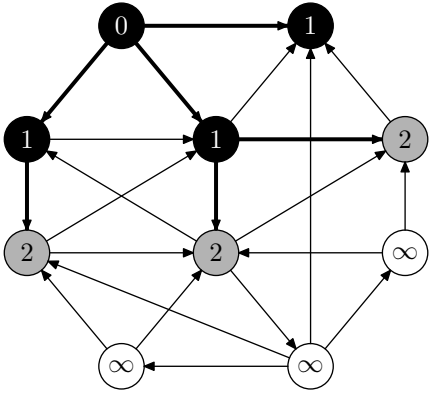
Breitensuche – Größeres Beispiel



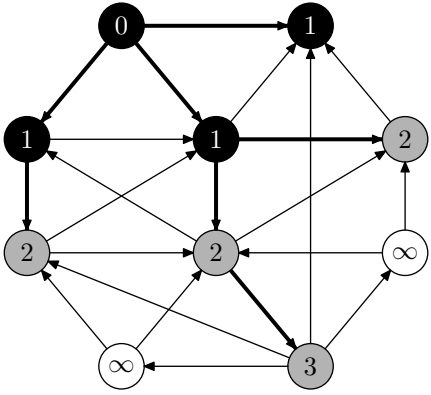
Breitensuche – Größeres Beispiel



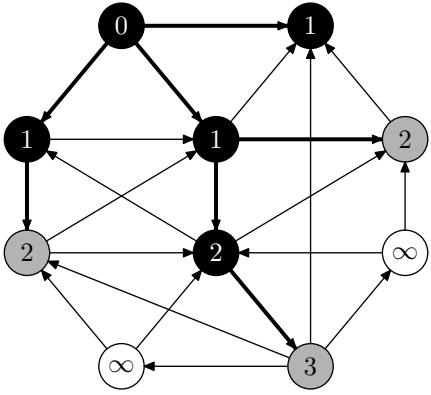
Breitensuche – Größeres Beispiel



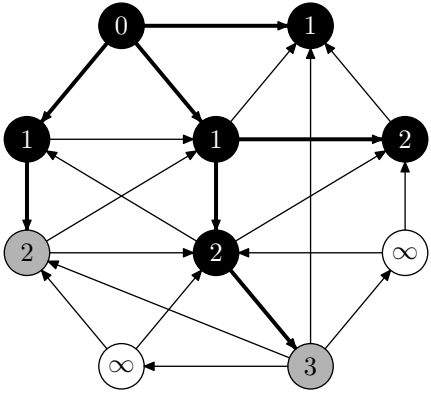
Breitensuche – Größeres Beispiel



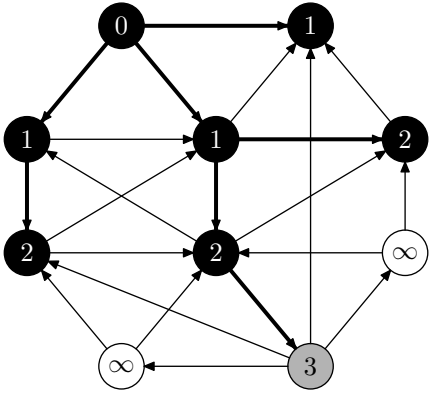
Breitensuche – Größeres Beispiel



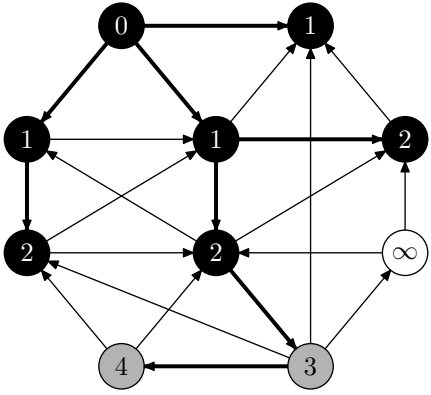
Breitensuche – Größeres Beispiel



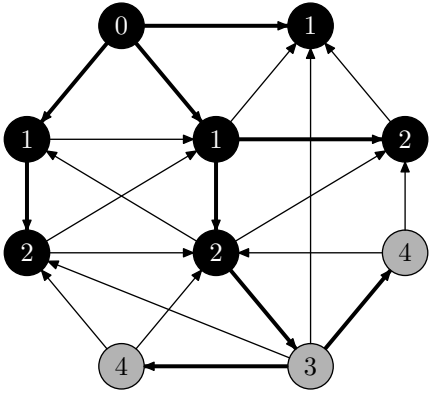
Breitensuche – Größeres Beispiel



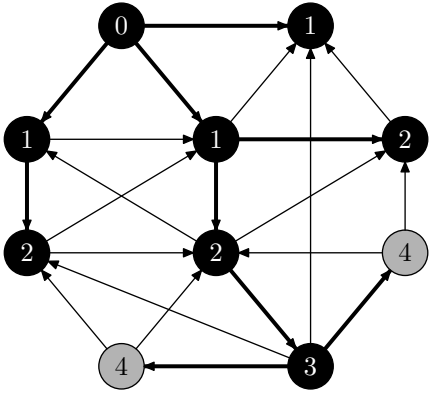
Breitensuche – Größeres Beispiel



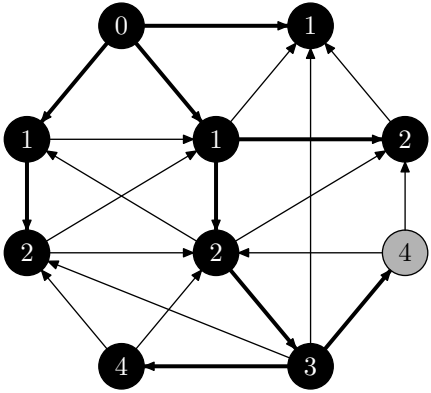
Breitensuche – Größeres Beispiel



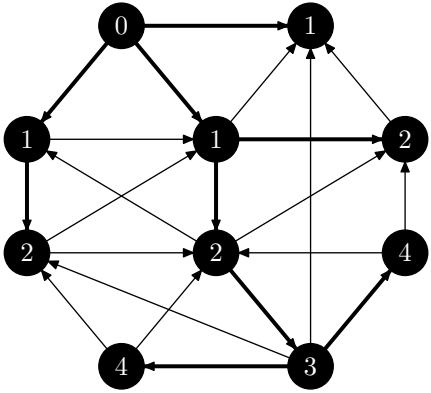
Breitensuche – Größeres Beispiel



Breitensuche – Größeres Beispiel



Breitensuche – Größeres Beispiel



Java

```
public static⟨V⟩ Map⟨V, V⟩ BFS(Graph⟨V⟩ G, V s) {
  Map⟨V, V⟩ pred = new HashMap⟨V, V⟩();
  Map⟨V, Integer⟩ color = new HashMap⟨V, Integer⟩();
  for(V u : G.allNodes()) color.put(u, WHITE);
  Queue⟨V⟩ q = new Queue⟨V⟩();
  q.enqueue(s); color.put(s, GRAY);
  while(!q.isempty()) {
    V u = q.dequeue();
    for(V v : G.neighbors(u)) {
      if(color.get(v) == WHITE) {
        color.put(v, GRAY);
        q.enqueue(v);
        pred.put(v, u);
      }
    }
    color.put(u, BLACK);
  }
  return pred;
}
```

Breitensuche

Theorem

Gegeben sei ein gerichteter oder ungerichteter Graph $G = (V, E)$ und ein Knoten $s \in V$.
Die kürzesten Wege von s zu allen anderen Knoten und die zugehörigen Abstände können in $O(|V| + |E|)$ berechnet werden.

Beweis.

Wir verwenden Breitensuche.
Die Laufzeit ist offensichtlich linear.

Der Korrektheitsbeweis sei hier weggelassen (etwas lang und technisch). □

Breitensuche

Theorem

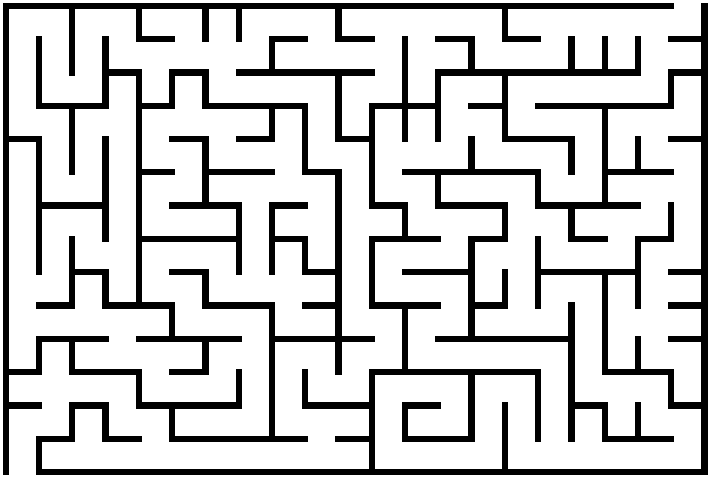
Gegeben sei ein gerichteter oder ungerichteter Graph $G = (V, E)$ und ein Knoten $s \in V$.
Die kürzesten Wege von s zu allen anderen Knoten und die zugehörigen Abstände können in $O(|V| + |E|)$ berechnet werden.

Beweis.

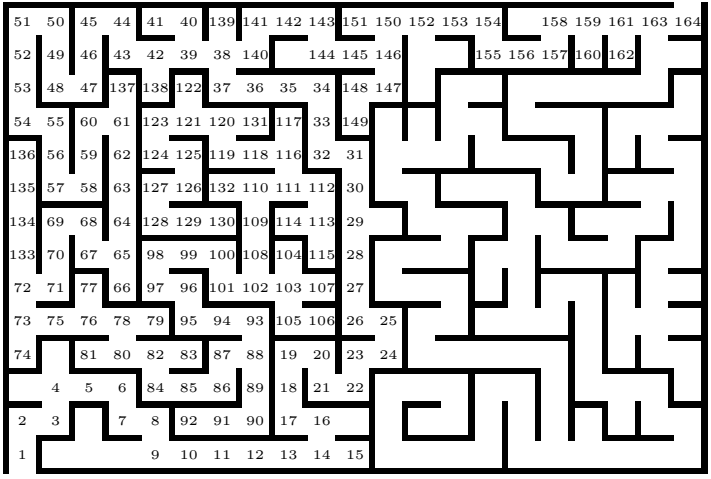
Wir verwenden Breitensuche.
Die Laufzeit ist offensichtlich linear.

Der Korrektheitsbeweis sei hier weggelassen (etwas lang und technisch). □

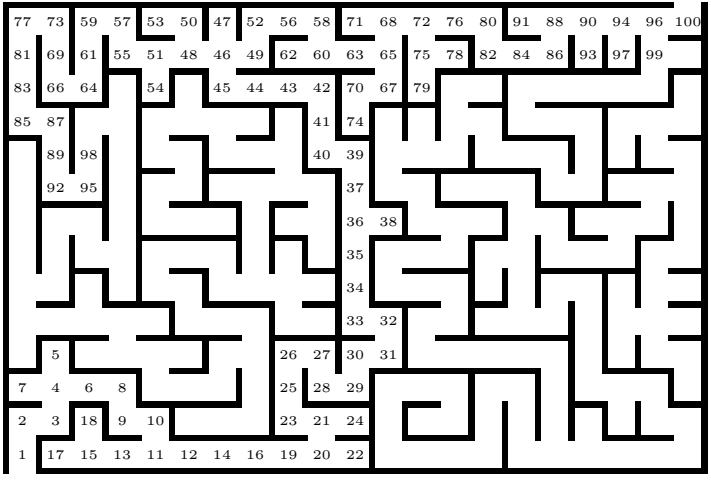
Beispiel



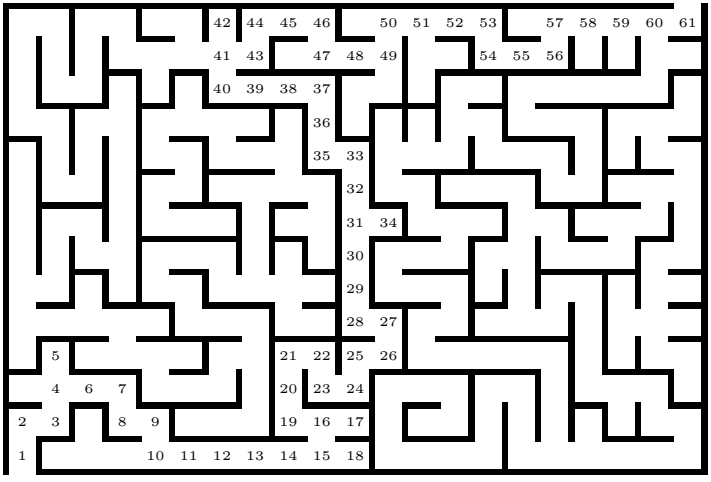
Beispiel: DFS



Beispiel: BFS



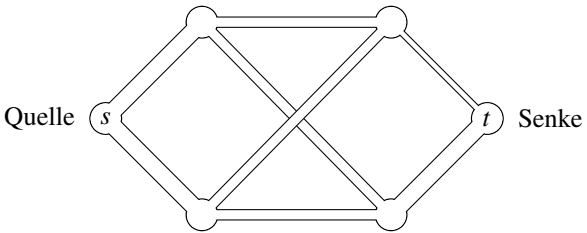
Beispiel: LC



Übersicht

Netzwerkfluß

Gegeben ist ein System von Wasserrohren:

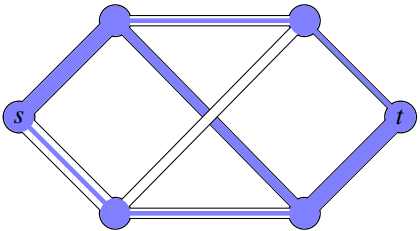


Die Kapazität jedes Rohres ist 3, 5 oder 8 l/s.

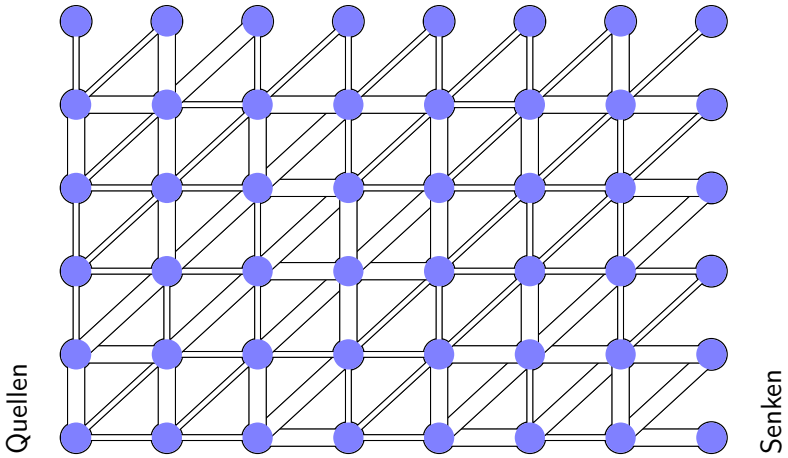
Frage: Wieviel Wasser kann von der Quelle zur Senke fließen?

Netzwerkfluß

Antwort: Maximal 11 l/s sind möglich.

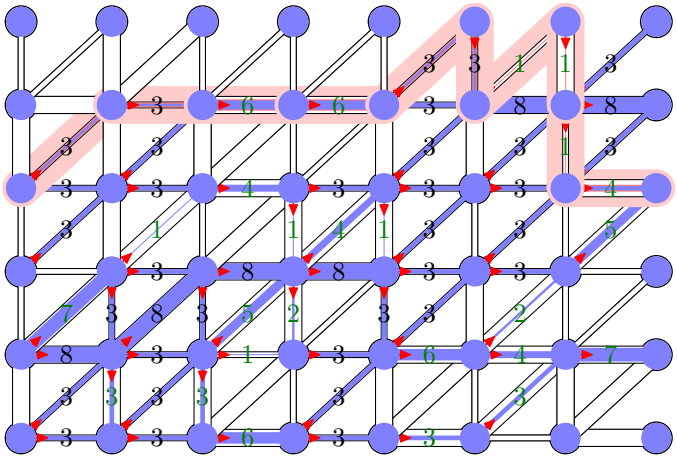


Netzwerkfluß – Aufgabe



Die Kapazitäten sind 3 und 8.

Netzwerkfluß – Lösung



Der maximale Fluß beträgt 30.

s-t-Netzwerke

Definition

Ein s-t-Netzwerk (flow network) ist ein gerichteter Graph $G = (V, E)$, wobei

1. jede Kante $(u, v) \in E$ eine Kapazität $c(u, v) \geq 0$ hat,
2. es eine Quelle $s \in V$ und eine Senke $t \in V$ gibt.

Es ist bequem, anzunehmen daß jeder Knoten auf einem Pfad von s nach t liegt.

Falls $(u, v) \notin E$ setzen wir $c(u, v) = 0$.

Es kann Kanten (u, v) und (v, u) mit verschiedener Kapazität geben.

s-t-Netzwerke

Definition

Ein s-t-Netzwerk (flow network) ist ein gerichteter Graph $G = (V, E)$, wobei

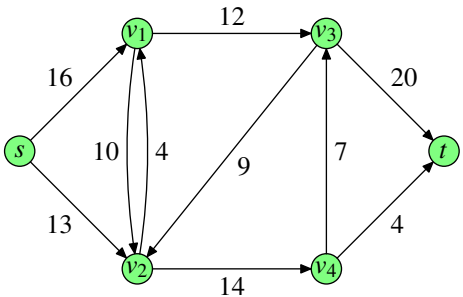
1. jede Kante $(u, v) \in E$ eine Kapazität $c(u, v) \geq 0$ hat,
2. es eine Quelle $s \in V$ und eine Senke $t \in V$ gibt.

Es ist bequem, anzunehmen daß jeder Knoten auf einem Pfad von s nach t liegt.

Falls $(u, v) \notin E$ setzen wir $c(u, v) = 0$.

Es kann Kanten (u, v) und (v, u) mit verschiedener Kapazität geben.

Beispiel eines s-t-Netzwerks



Die Kanten sind mit den Kapazitäten $c(u, v)$ beschriftet.

Flüsse

Definition

Ein Fluß ist eine Funktion $f : V \times V \to \mathbf{R}$, die Paare von Knoten auf reelle Zahlen abbildet und diese Bedingungen erfüllt:

- Zulässigkeit: Für $u, v \in V$ gilt $f(u, v) \leq c(u, v)$.
- Symmetrie: Für $u, v \in V$ gilt $f(u, v) = -f(v, u)$.
- Flußerhaltung: Für $u \in V - \{s, t\}$ gilt $\sum_{v \in V} f(u, v) = 0$.

Der Wert $|f|$ eines Flusses ist definiert als $|f| = \sum_{u \in V} f(s, u)$.

Dies ist gerade der Gesamtfluß aus der Quelle heraus.

Maximale Flüsse

Das Problem des maximalen Flusses:

Gegeben: Ein s-t-Netzwerk.

Gesucht: Ein Fluß mit maximalem Wert.

Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

Maximale Flüsse

Das Problem des maximalen Flusses:

Gegeben: Ein s-t-Netzwerk.

Gesucht: Ein Fluß mit maximalem Wert.

Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

Maximale Flüsse

Das Problem des maximalen Flusses:

Gegeben: Ein s-t-Netzwerk.

Gesucht: Ein Fluß mit maximalem Wert.

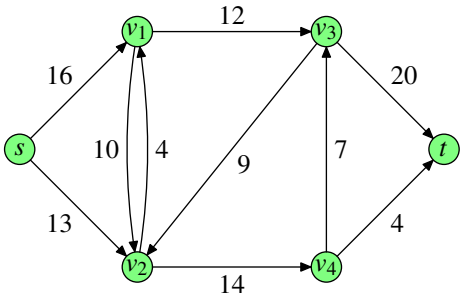
Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

- Weltraumtouristen machen Angebote
- Sie benötigen spezielle Ausrüstung
- Ausrüstung kann mehrfach benutzt werden
- Wer soll mitgenommen werden?

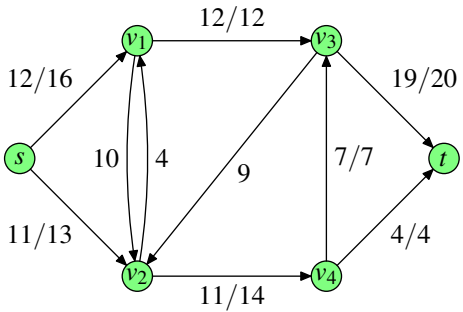
Was ist der maximale Fluß?



Der maximale Fluß ist 23.

Die Kanten sind mit $c(u, v)$ beschriftet oder mit $f(u, v)/c(u, v)$, falls $f(u, v) > 0$.

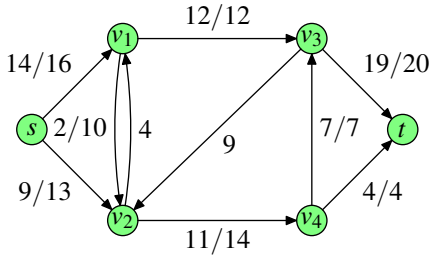
Was ist der maximale Fluß?

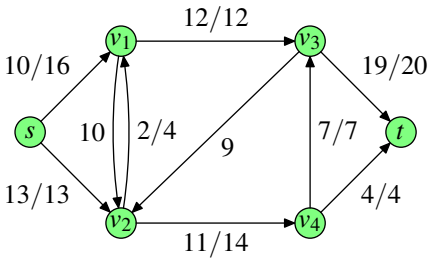


Der maximale Fluß ist 23.

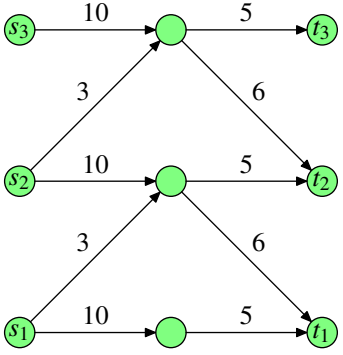
Die Kanten sind mit $c(u, v)$ beschriftet oder mit $f(u, v)/c(u, v)$, falls $f(u, v) > 0$.

Andere optimale Lösungen



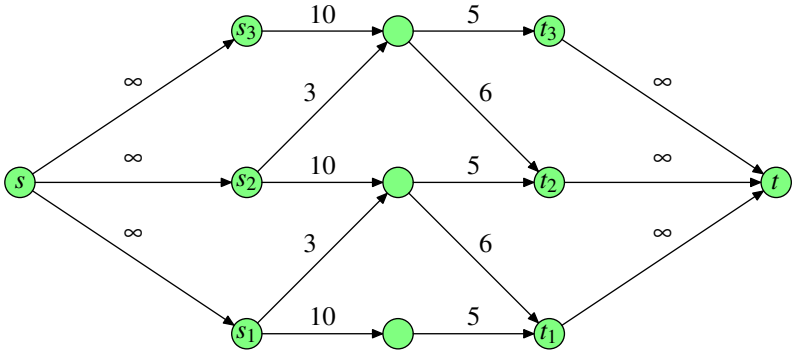


Mehrfache Quellen oder Senken



Mehrfache Quellen oder Senken sind eine Verallgemeinerung des Problem des maximalen Flusses.

Mehrfache Quellen oder Senken



$\rightarrow$ Neue „Superquelle“ und „Supersenke“ hinzufügen.

Existenz des maximalen Flusses

Existiert stets der maximale Fluß

$$\max\left\{ |f| \;\middle|\; f \text{ ist ein } s\text{-}t\text{-Fluß in } G \right\}?$$

Ja, denn die Menge aller Flüsse ist abgeschlossen im $\mathbf{R}^m$ und sie ist nicht leer.

Die stetige Funktion, die einen Fluß auf ihren Wert abbildet, hat daher ein Maximum:

$$|f| = \sum_{u \in V} f(s,u) \text{ ist stetig!}$$

(Satz von Weierstrass)

Existenz des maximalen Flusses

Existiert stets der maximale Fluß

$$\max\{\, |f| \mid f \text{ ist ein } s\text{-}t\text{-Fluß in } G \,\}?$$

Ja, denn die Menge aller Flüsse ist abgeschlossen im $\mathbf{R}^m$ und sie ist nicht leer.

Die stetige Funktion, die einen Fluß auf ihren Wert abbildet, hat daher ein Maximum:

$$|f| = \sum_{u \in V} f(s,u) \text{ ist stetig!}$$

(Satz von Weierstrass)