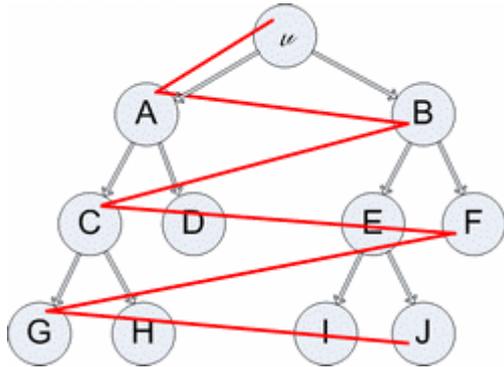


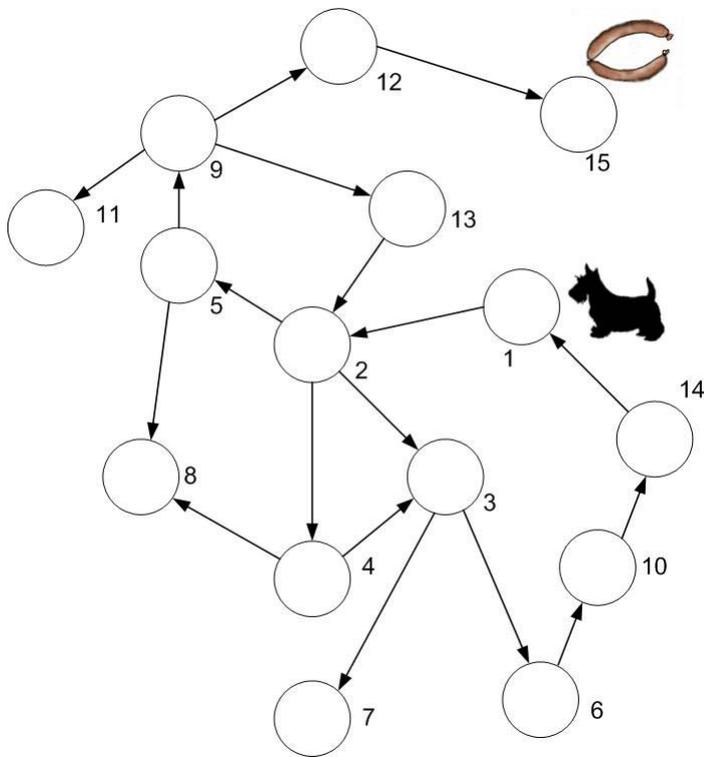
Informatik OTG	Graphen	
	Breitensuche	

### Arbeitsweise



Man geht vom Startknoten  $v$  aus und besucht zunächst alle Knoten, die den Abstand  $k = 1$  von  $v$  haben. Sind diese Knoten in einer Warteschlange gespeichert, werden alle direkten Nachbarn dieser Knoten besucht, nicht entdeckte Knoten kommen in die Warteschlange. Die nächste Tiefenebene wird also erst dann begonnen, wenn alle Knoten der momentanen Ebene schon besucht sind.

Die Front zwischen entdeckten und unentdeckten Knoten erstreckt sich gleichmäßig über die gesamte Breite, daher kommt der Name Breitensuche.



### Algorithmus

1. Bestimme den Knoten, an dem die Suche beginnen soll, und speichere ihn in einer Warteschlange ab.
2. Entnimm einen Knoten vom Beginn der Warteschlange und markiere ihn.
3. Falls das gesuchte Element gefunden wurde, brich die Suche ab und liefere "gefunden" zurück.
4. Ansonsten hänge alle bisher unmarkierten Nachfolger dieses Knotens, die sich noch nicht in der Warteschlange befinden, ans Ende der Warteschlange an.
5. Wiederhole ab Schritt 2.
6. Ist die Warteschlange leer, dann wurde jeder Knoten bereits untersucht. Beende die Suche und liefere "nicht gefunden" zurück.

**Was ist eine Warteschlange?** Eine Warteschlange, auch Queue bezeichnet, ist eine sich dynamisch ändernde Datenstruktur, in welche man Daten speichern und aus der man Daten entfernen kann. Das Entfernen der Elemente erfolgt nach dem FIFO - Prinzip, d.h. das am längsten in der Warteschlange verweilende Element wird als erstes entfernt.

### Speicherplatzverbrauch

$O$  (alle Kanten + alle Knoten)

### Laufzeit

$O$  (alle Kanten + alle Knoten), weil jeder Knoten einmal auf den Speicher gelegt wird und jede Kante Speicherplatzverbrauch

### Anwendung

- alle Zusammenhangskomponenten in einem Graph finden
- alle Knoten innerhalb einer Zusammenhangskomponente finden
- den kürzesten Pfad zwischen zwei Knoten finden
- das Kürzeste-Kreise-Problem