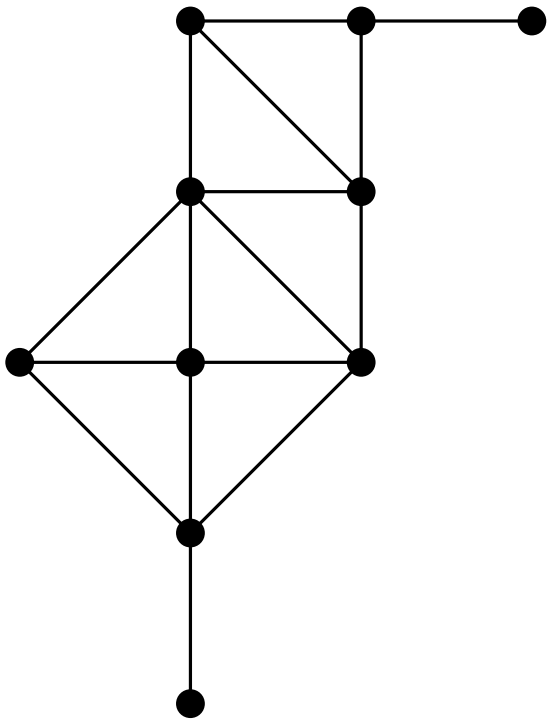
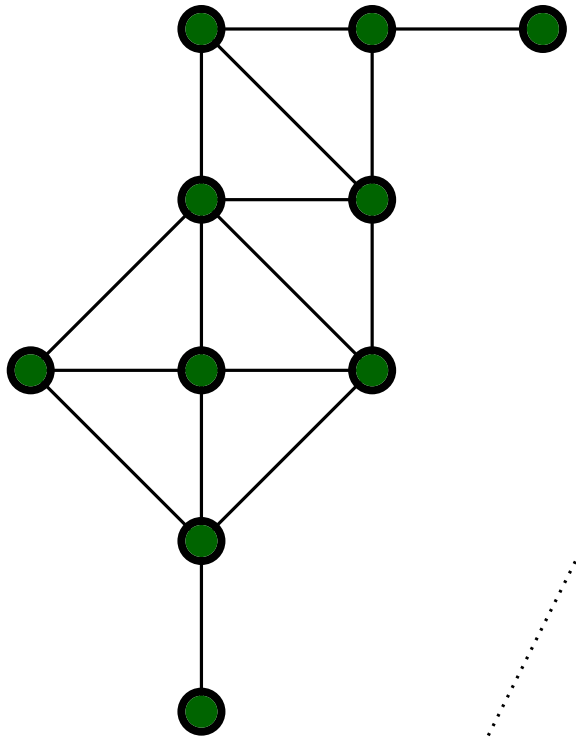


Algorithmus **LexBFS**

---

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende



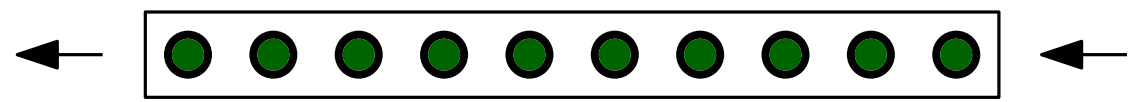
Algorithmus **LexBFS**

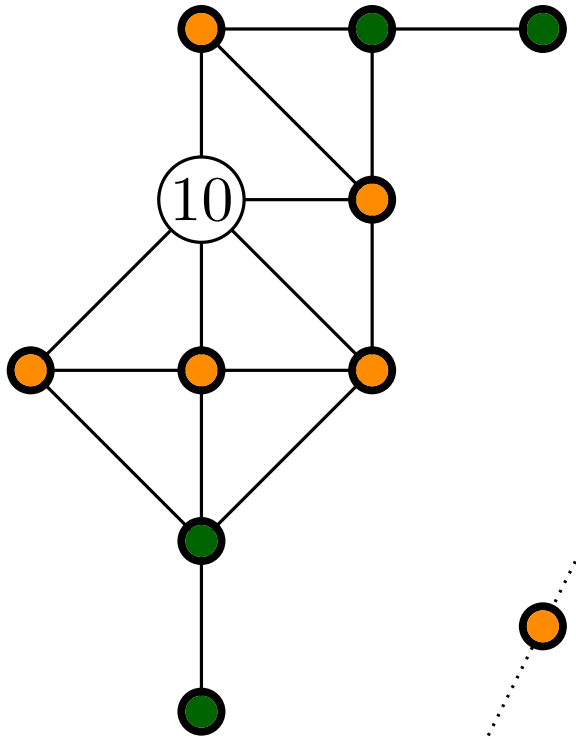
für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= \emptyset$

Queue





Algorithmus **LexBFS**

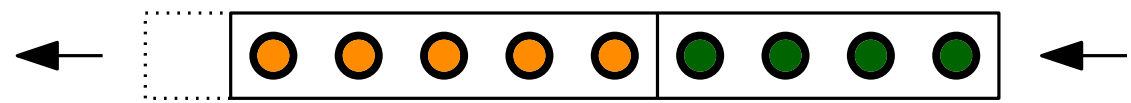
für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

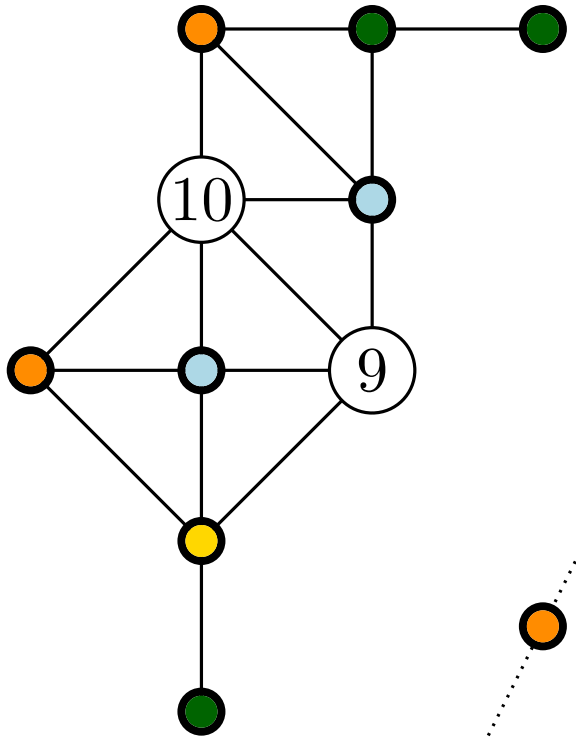
$= 10$

Labels

$= \emptyset$

Queue





Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

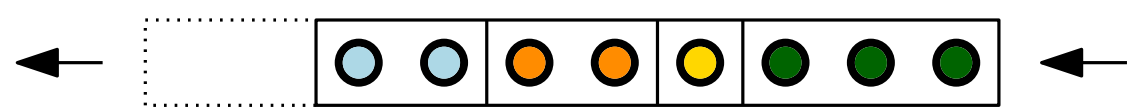
$= 10.9$

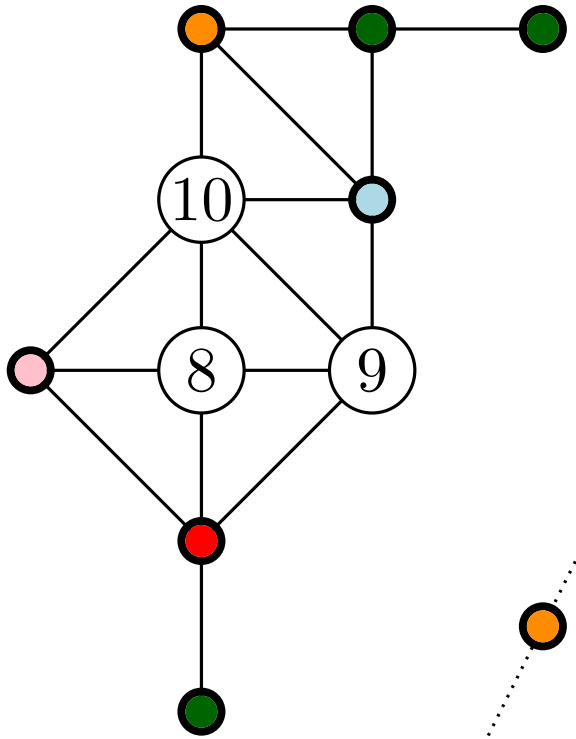
$= 10$

$= 9$

$= \emptyset$

Queue





Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$

$= 10.8$

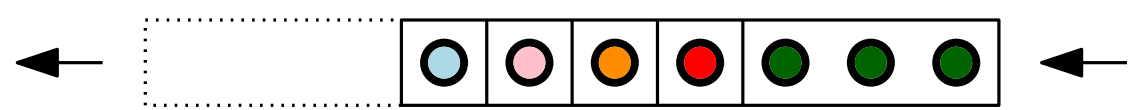
$= 10$

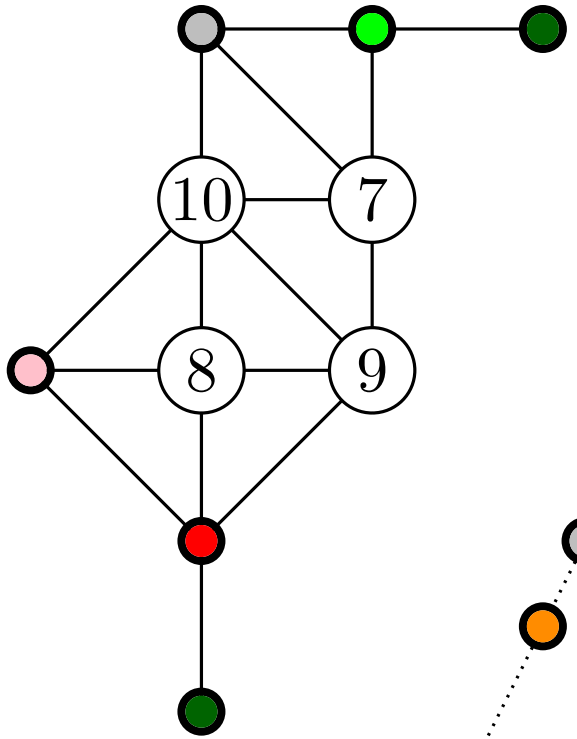
$= 9.8$

$= 9$

$= \emptyset$

Queue





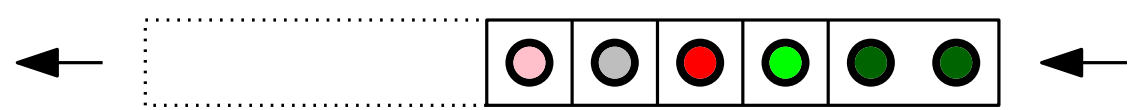
Algorithmus **LexBFS**

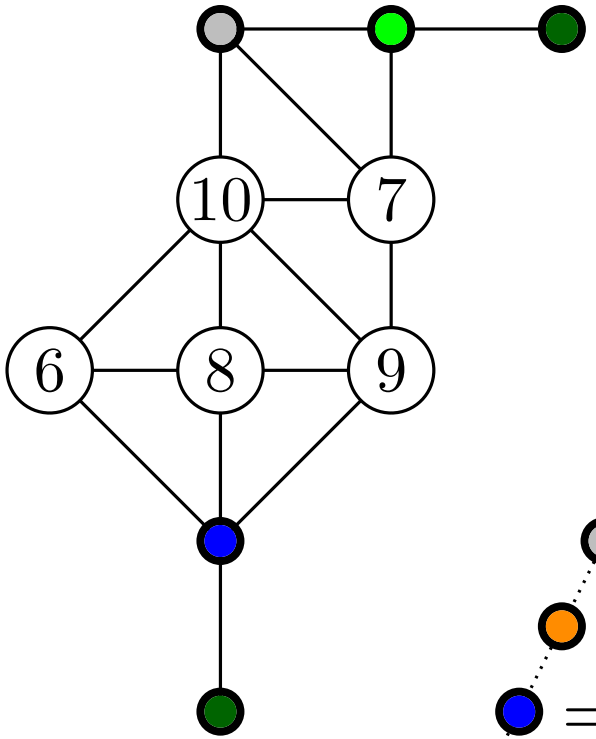
für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8$
$= 9$
$= 7$
$= \emptyset$

Queue





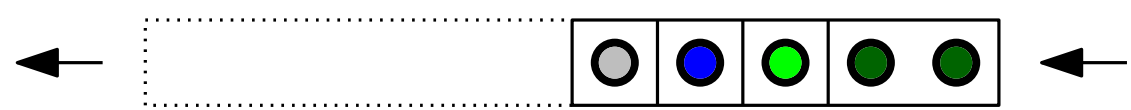
Algorithmus **LexBFS**

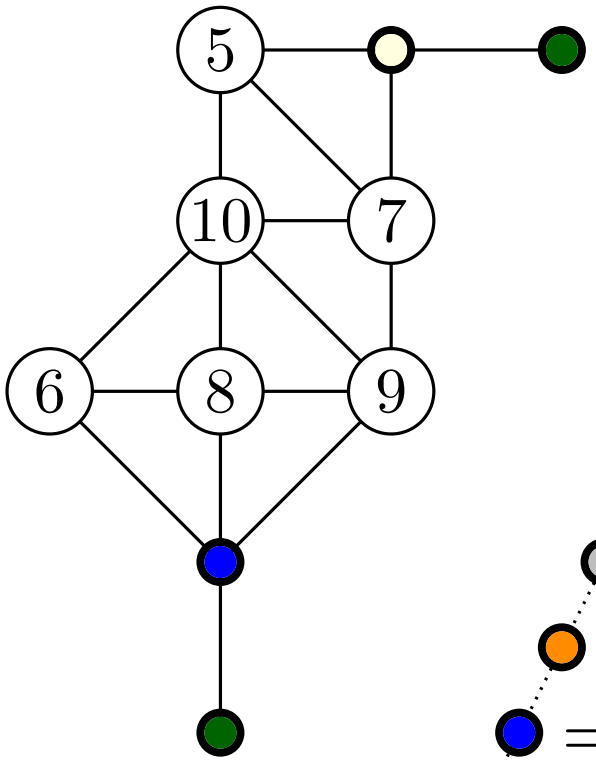
für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7$
$= \emptyset$

Queue





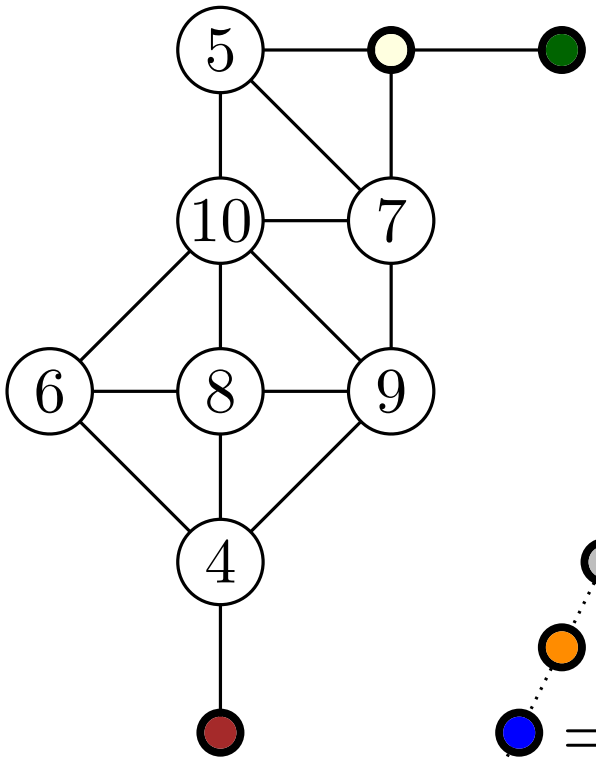
Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7.5$
$= 7$
$= \emptyset$

Queue



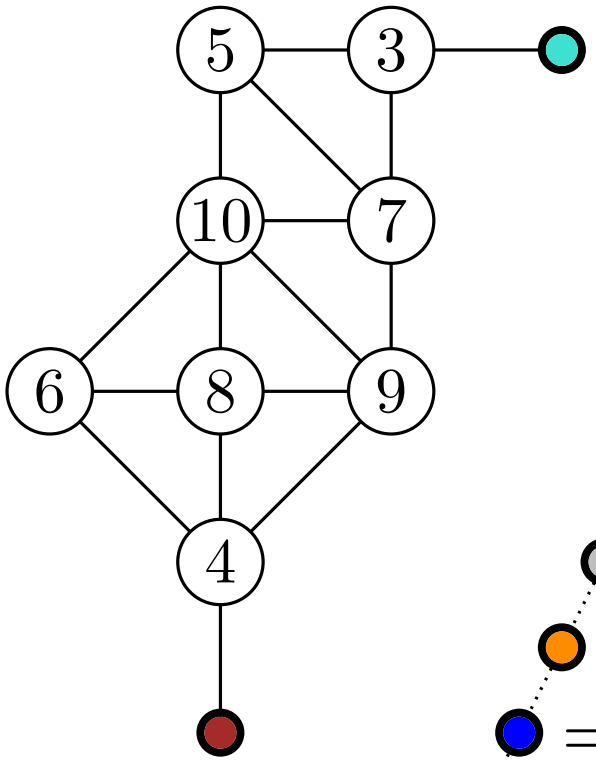
Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7.5$
$= 7$
$= 4$
$= \emptyset$

Queue



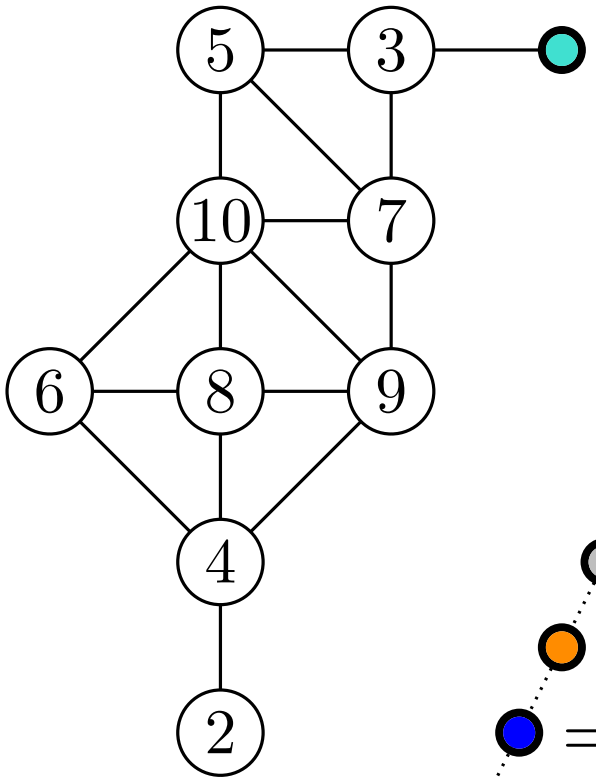
Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7.5$
$= 7$
$= 4$
$= 3$
$= \emptyset$

Queue



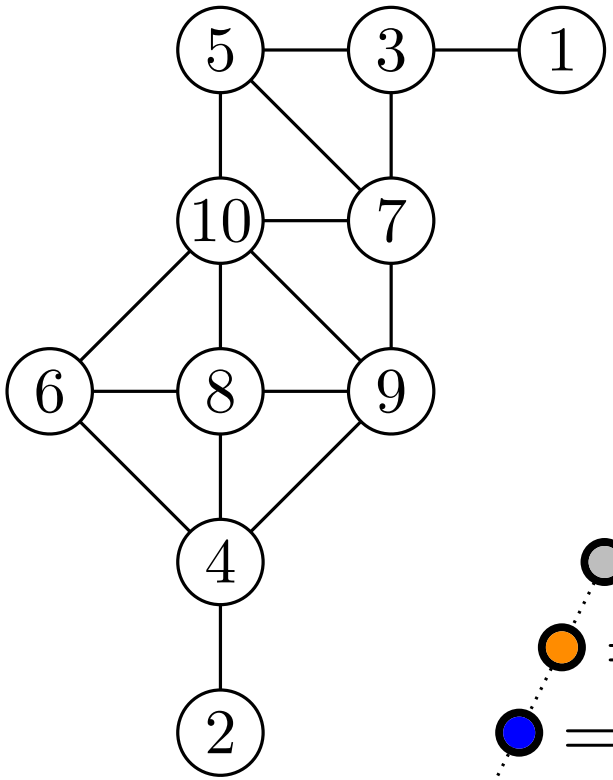
Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7.5$
$= 7$
$= 4$
$= 3$
$= \emptyset$

Queue



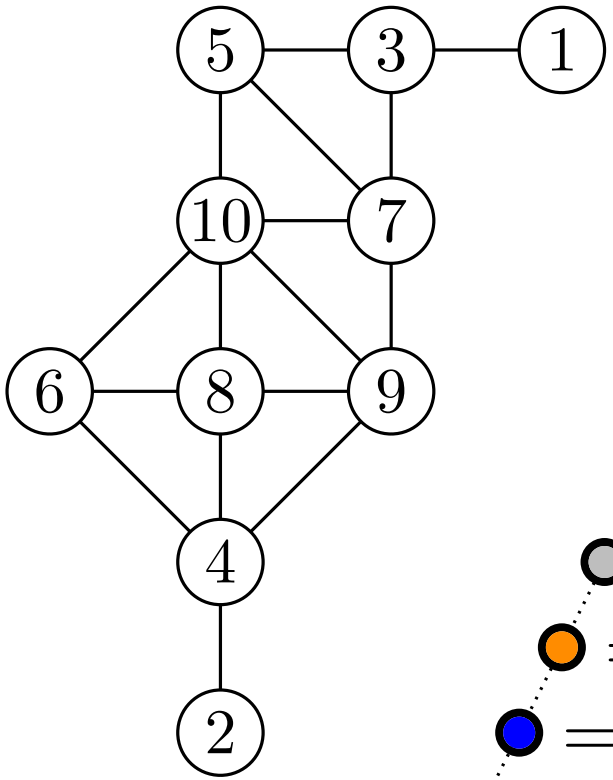
Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende

Labels

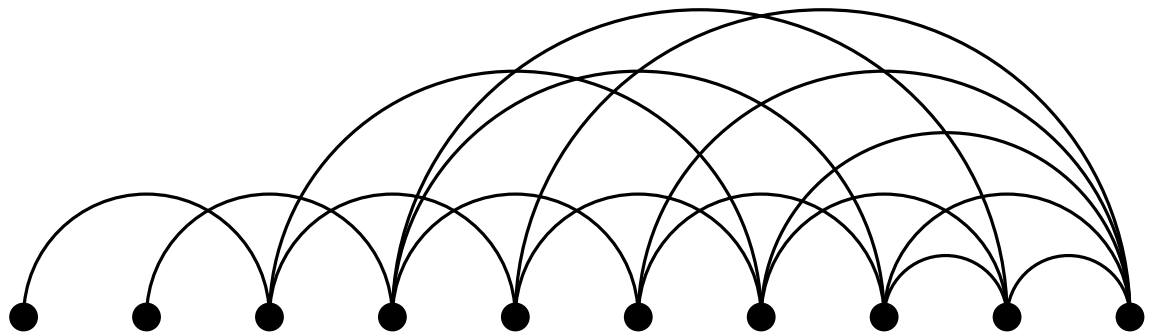
$= 10.9$
$= 10.8$
$= 10.7$
$= 10$
$= 9.8.6$
$= 9.8$
$= 9$
$= 7.5$
$= 7$
$= 4$
$= 3$
$= \emptyset$

Queue



Algorithmus **LexBFS**

für $i \leftarrow n$ **bis** 1 **tue**
  $v \leftarrow$ mit größtem Label;
  $\sigma(i) \leftarrow v$;
  für $w \in \mathrm{Adj}(v)$ **tue**
    $\mathrm{Label}(w) \leftarrow \mathrm{Label}(w).i$;
  Ende
Ende



```
1  für w ∈ Adj(v) nicht nummeriert tue
2      wenn Flag(Set(w)) = false dann
3          Füge neue Menge S hinter Set(w) in Q ein;
4          Flag(Set(w)) ← true; Füge S in FixList ein;
5      Ende
6      S ← Menge hinter Set(w) in Q;
7      Entferne w aus Set(w); Füge w in S ein;
8      Set(w) ← S;
9  Ende
10 für S ∈ FixList tue
11     Flag(S) ← false;
12     wenn S leer dann
13         Entferne S aus Q;
14     Ende
15     Entferne S aus FixList;
16 Ende
```

Algorithmus 2 : Updateschritt in LexBFS

Eingabe : Graph $G = (V, E)$, Knotenordnung σ.
Ausgabe : Wahr, wenn σ PES, sonst falsch.

```
1  für alle Knoten v tue A(v) ← ∅;
2  für i ← 1 bis n − 1 tue
3  |  v ← σ(i);
4  |  X ← {x ∈ Adj(v) | σ(v) < σ(x)};
5  |  wenn X = ∅ dann gehe zu Zeile 8;
6  |  u ← argmin{σ(x) | x ∈ X};
7  |  Füge X − {u} in A(u) ein;
8  |  wenn A(v) − Adj(v) ≠ ∅ dann
9  |  |  return falsch;
10 |  Ende
11 Ende
12 return wahr;
```

Algorithmus 3 : Test auf perfektes Eliminationsschema

Eingabe : Listen $\mathrm{Adj}(v)$, $A(v)$.
Ausgabe : Wahr, wenn $A(v) - \mathrm{Adj}(v) \neq \emptyset$, sonst falsch.

```
1  für w ∈ Adj(v) tue Test(w) ← true;
2  für w ∈ A(v) tue
3  |  wenn Test(w) = false dann
4  |  |  return wahr;
5  |  Ende
6  Ende
7  für w ∈ Adj(v) tue Test(w) ← false;
8  return falsch;
```

Algorithmus 4 : Test auf $A(v) - \mathrm{Adj}(v) \neq \emptyset$ in Zeile **8**