Eingabe : Ungerichteter Graph $G = (V, E)$.
Ausgabe : Knotenordnung σ.

```
1  Weise jedem Knoten das Label ∅ zu;
2  für i ← n bis 1 tue
3  |  wähle einen nicht nummerierten Knoten v
   |                                  mit größtem Label;
4  |  σ(i) ← v;
5  |  für jeden nicht nummerierten Knoten w ∈ Adj(v)
   |  |  füge i zu Label(w) hinzu;
6  |  Ende
7  Ende
```

Algorithmus 1 : LexBFS

```
1  für w ∈ Adj(v) nicht nummeriert tue
2      wenn Flag(Set(w)) = false dann
3          Füge neue Menge S vor Set(w) in Q ein;
4          Flag(Set(w)) ← true; Füge Set(w) in FixList ein;
5      Ende
6      S ← Menge vor 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.

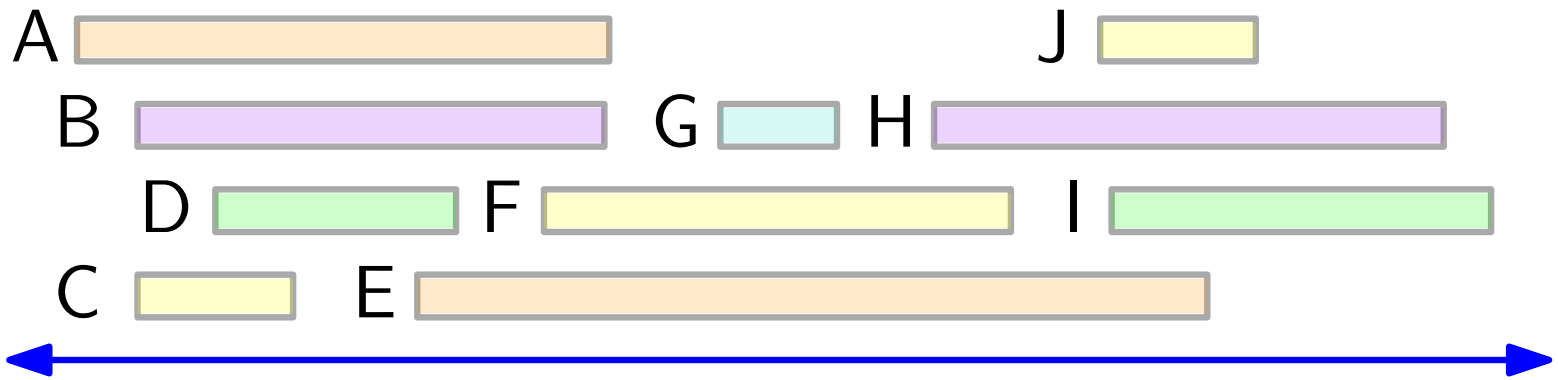
```
1  für alle Knoten v tue A(v) ← ∅;
2  für i ← 1 bis n − 1 tue
3  |  v ← σ(i);
4  |  X_v ← Adj(v) ∩ {σ(i + 1), . . . , σ(n)};
5  |  wenn X_v = ∅ dann gehe zu Zeile 8;
6  |  u ← argmin{σ(x) | x ∈ X_v};
7  |  Füge X_v − {u} in A(u) ein;
8  |  wenn A(v) − Adj(v) ≠ ∅ dann
9  |  |  return falsch;
10 |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

Algorithmus 3 : Test auf perfektes Eliminationsschema

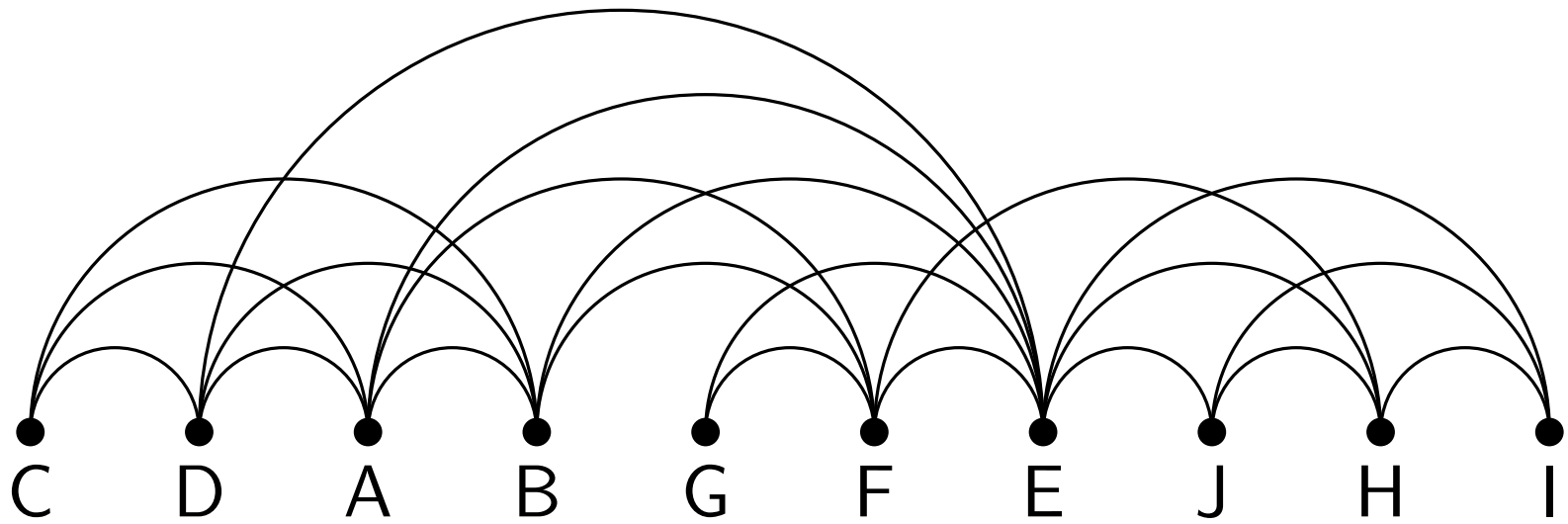
Eingabe : Listen $\mathrm{Adj}(v)$, $A(v)$.
Ausgabe : Wahr, wenn $A(v) - \mathrm{Adj}(v) \neq \emptyset$, sonst falsch.

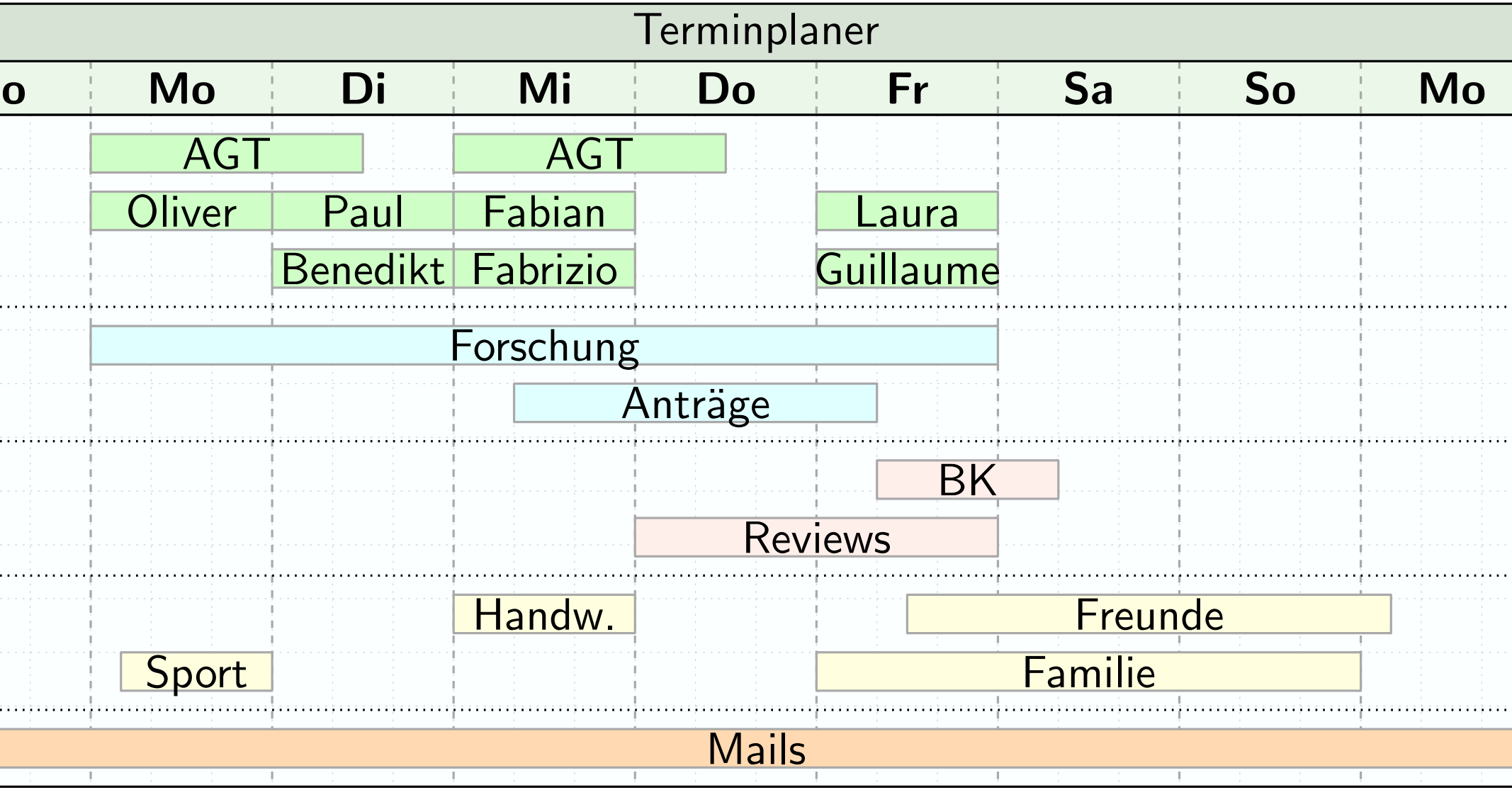
```
1  für w ∈ Adj(v) tue Test(w) ← true;
2  für w ∈ A(v) tue
3  |  wenn Test(w) = false dann
4  |  |  return wahr;
5  |  Ende
6  Ende
7  für w ∈ Adj(v) tue Test(w) ← false;
8  return falsch;
```

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



Für **Intervallgraphen** liefert die **links-nach-rechts Ordnung der rechten Endpunkte** in einer Intervall-Repräsentation ein **PES**.

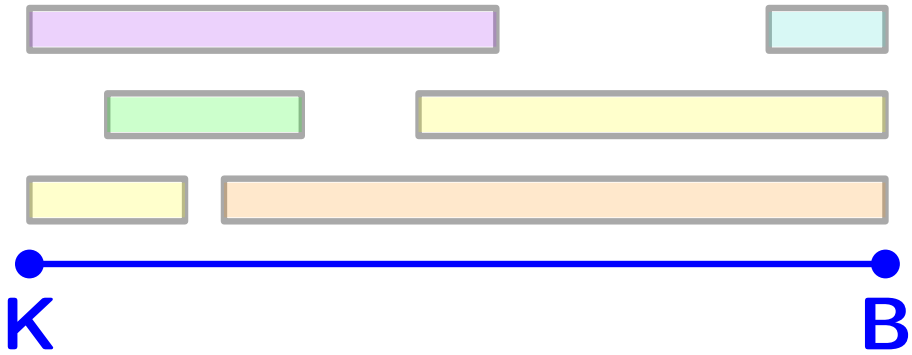


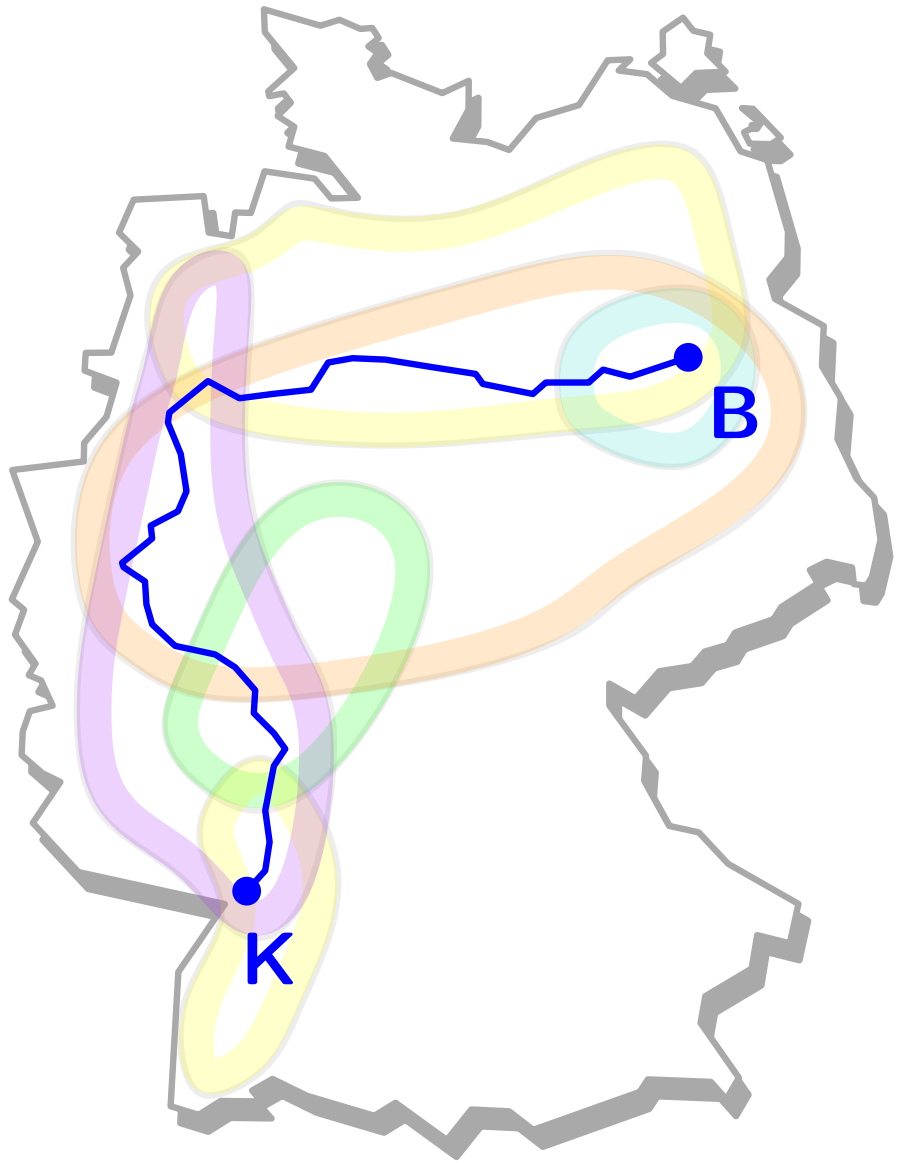


In wieviele Personen müsste ich mich aufteilen, um alles zu schaffen? $\chi(G)$

Wieviel kann ich alleine schaffen? $\alpha(G)$

- Ich reise von **K** nach **B** durch Deutschlands Regionen





Wie oft müsste ich anhalten, um alle Regionen zu sehen?

$$k(G)$$

Wieviele Regionen kann ich gleichzeitig sehen?

$$\omega(G)$$

Eingabe : chordaler Graph $G = (V, E)$.
Ausgabe : Clique C und Knotenfärbung ϕ.

1 Bestimme mit LexBFS ein PES σ von G;
2 $C \leftarrow \emptyset$, $\phi \leftarrow 0$;
3 **für** $i \leftarrow n$ **bis** 1 **tue**
4 $\quad v \leftarrow \sigma(i)$;
5 $\quad X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \ldots, \sigma(n)\}$;
6 $\quad \phi(v) \leftarrow \min(\mathbb{N} - \{\phi(w) \mid w \in X_v\})$;
7 $\quad$ **Wenn** $|C| < |X_v + \{v\}|$, **dann**
8 $\quad\quad C \leftarrow X_v + \{v\}$;
9 $\quad$ **Ende**
10 **Ende**
11 **Gebe aus** ϕ und C;

Algorithmus 5 : Bestimmung von $\omega(G)$ und $\chi(G)$