Algorithmus

Fall: Max. $\lambda - 1$ Friendly-Knoten

1. Improved-Graph G' berechnen
   $\rightarrow \exists$ I.simp.-Knoten v mit $\deg(v) = k + 1 \Rightarrow$ STOP
2. Alle I.simp.-Knoten in Menge SL und von G entfernen $\Rightarrow \widehat{G}$ entsteht
   $\rightarrow |SL| < c_2 \cdot |V| \Rightarrow$ STOP (THEOREM 4.2.)
3. Algorithmus rekursiv auf $\widehat{G}$ ausführen $\Rightarrow$ Ausgabe von Zerteilung (Y, T) von $\widehat{G}$
   $\rightarrow \mathrm{tw}(\widehat{G}) > k \Rightarrow$ STOP ($\widehat{G}$ Teilgraph von $G \Rightarrow \mathrm{tw}(G) > k$)
4. $\forall v \in SL$: Finde ein $Y_{i_v} \in Y$ in dem alle Nachbarn von v sind $(N_G(v) \subseteq Y_{i_v})$
5. Füge $Y_{j_v} = \{v\} \cup N_G(v)$ zu Y hinzu und mache es adjazent zu Y_{i_v}
   $\rightarrow$ Baumzerteilung von G mit Baumweite max. k

Algorithmus

Fall: Max. $\lambda - 1$ Friendly-Knoten: Letzter Schritt

1. $\forall v \in SL$: Finde ein $Y_{i_v} \in Y$ in dem alle Nachbarn von v sind ($N_G(v) \subseteq Y_{i_v}$)
2. Füge $Y_{j_v} = \{\{v\} \cup N_G(v)\}$ zu Y hinzu und mache es adjazent zu Y_{i_v}
   $\Rightarrow$ Baumzerteilung von G mit Baumweite max. k

Y_{i_v} existiert für jedes v, da I.simp.-Knoten in G nicht adjazent sind und $N_G(v)$ eine Clique formt.
LEMMA 2.1.i): "Ist (X, T) Zerteilung von G und formt $W \subseteq V$ eine Clique in G, $\exists i \in IW \subseteq X_i$"

Laufzeitanalyse

Fall 1

1. Improved-Graph G' berechnen