<u>Lemma 2.14</u> Sei $G = (V, E)$ ein (Multi-)Graph und $u, v \in V$. Dann gilt:

$$\lambda(G) = \min\{\text{cut}(u, v), \lambda(G_{u,v})\}.$$

<u>Def:</u> Sei $G = (V, E)$ ein (Multi-)Graph. Eine Ordnung $v_1, \ldots, v_n$ der Knoten heißt *adjazenzmaximal*, falls

$$e(v_i, \{v_1, \ldots, v_{i-1}\}) \geq e(v_j, \{v_1, \ldots, v_{i-1}\})$$

für alle $1 \leq i < j \leq n$.

<u>Lemma 2.15</u> Sei $G = (V, E)$ ein (Multi-)Graph und $v_1, \ldots, v_n$ eine adjazenzmaximale Ordnung. Dann gilt:

$$\text{cut}(v_n, v_{n-1}) = e(v_n, \{v_1, \ldots, v_{n-1}\}).$$

Algorithmus $\textsc{MinCut}(G)$

$\textsc{Eingabe}$: Graph $G = (V, E)$

$\textsc{Ausgabe}$: Kantenzusammenhangszahl $\lambda(G)$

//Initialisierung

$\lambda := \min\{\text{deg}(v) \mid v \in V\}$; // Kandidat für $\lambda(G)$

//Schleife

for $k := n$ **downto** 2 **do**

  Berechne adjazenzmaximale Ordnung $v_1, \ldots, v_k$ von G;

  if $e(v_k, \{v_1, \ldots, v_{k-1}\}) < \lambda$ **then**

    $\lambda := e(v_k, \{v_1, \ldots, v_{k-1}\})$;

  $G := G_{v_k\, v_{k-1}}$; // verschmelze v_k und v_{k-1}