Bipartites Matching

Eingabe: zshgd, bipartiter Graph $G = (A \cup B, E)$

Ausgabe: maximales Matching $M \subseteq E$

$M := \emptyset$;

`loop:`

for all $v \in A \cup B$ **do** label[v] := 0;

// beginne BFS bei allen ungematchten Knoten von A

$R := \emptyset; l := 1;$

for all $v \in A$, v ungematcht **do**

for all $\{v, w\} \in E$ **do**

label$[w] := l; R := R \cup \{w\};$

while R $\neq \emptyset$ **and** R enthält keinen ungematchten Knoten **do**

// folge gematchten Kanten aus R zu Knoten aus A

$L := \emptyset; l := l + 1;$

for all $w \in R$, $\{v, w\} \in$ **do**

$L := L \cup \{v\}$; label$[v] := l$;

// folge ungematchten Kanten aus L zu Knoten aus B

$R := \emptyset; l := l + 1;$

for all $v \in L$, $\{v, w\} \in E \setminus M$ **do**

if label$[w] = 0$ **then**

$R := R \cup \{w\}$; label$[w] := l$;

if R = $\emptyset$ **then** STOP; // M ist maximales Matching;

⋮

⋮

// Starte bei jedem ungematchten Knoten aus A einen DFS

Markiere alle Knoten als “unchecked”;

for all $v \in A$, v ungematcht **do**

  Suche augmentierenden Pfad P Länge $\leq l$ mit

    Startknoten v, betrachte dabei nur Knoten, die als “unchecked” markiert sind und gehe DFS-mäßig vor; markiere alle bei der Suche betrachteten Knoten als “checked”;

  if Falls ein Pfad P gefunden wurde **then**

    $M := M \oplus P$; // augmentiere entlang P

goto `loop`;