

# 13. Sonderübung im Sommersemester 2020

Nadine Schärmann & Thomas Klütz

## Hamilton-Kreis: Backtracking

In einem Graphen  $G = (V, E)$ , ist  $V$  die Menge der *Ecken* („vertices“) und  $E \subseteq V \times V$  die Menge der *Kanten* („edges“). Ein *Weg* in  $G$  ist eine Folge  $(v_1, v_2, \dots, v_n)$  in  $V$  von Ecken, für die  $(v_i, v_{i+1}) \in E$ ,  $i = 1, \dots, n-1$ .

Ein *Hamilton-Weg* ist ein Weg, der jede Ecke von  $G$  genau einmal besucht. Ein *Hamilton-Kreis* ist ein Hamilton-Weg mit  $(v_n, v_1) \in E$ .

Der *Grad*  $\deg_G(v) = |\{w \in V | (v, w) \in E\}|$  einer Ecke ist die Anzahl der ausgehenden Kanten an der Ecke. Die Ecken  $w \in V$ , für die  $(v, w) \in E$ , heißen *Nachbarn* von  $v$ . Der *maximale Grad* eines Graphen sei  $\deg(G) = \max_{v \in V} \deg(v)$ .

Schreiben Sie ein Fortran95-Programm, was in einem beliebigen (gerichteten) Graphen mittels Backtracking einen Hamilton-Weg sucht.

Es kann vereinfachend angenommen werden, dass  $\deg(G)$  bekannt ist und mit  $d$  bezeichnet wird und es vertretbar ist, bei jeder Ecke  $d$  viele Speicherplätze für Nachbarecken bereit zu halten. Ebenso ist die Anzahl der Ecken  $|V|$  bekannt. Realisieren Sie einen Graphen in Fortran mithilfe einer Pointer-Struktur. Jede Ecke  $v \in V$  beinhaltet also einen Pointer pro Nachbar und eine ganzzahlige Komponente für den Schlüssel der Ecke.

Bei dem Backtracking-Algorithmus zum Finden eines Hamilton-Kreises wird mit einer beliebigen Ecke begonnen, alle Nachbarn nacheinander als zweite Ecke angenommen und der Algorithmus rekursiv ausgeführt, wobei die bereits gewählten Ecken nicht mehr genutzt werden. Sind alle Ecken besucht und  $v_1$  Nachbar von der letzten Ecke, wird der gefundene Hamilton-Kreis ausgegeben und der Algorithmus beendet. Es müssen also nicht alle Hamilton-Kreise gefunden werden.