

Programmierkurs Python II – SS 2010

Übung 3

1 Wikipedia-Graph mit Gewichten (2 Punkte)

Die Datei `wikipedia.txt` enthält eine kleine Sammlung von Wikipedia-Links. Die Links sind als Listen von String-Paaren gespeichert, wobei jeder String ein Seitentitel (im Wikipedia-Titel-Format) ist. Die Strings sind mit Tabulatoren getrennt; der erste markiert die Quelle, der zweite das Ziel eines Links.

Parse die Textdatei zu einem gerichteten Graphen, dessen Knoten mit den Wikipedia-Seitentiteln markiert sind und dessen Kanten Wikipedia-Links symbolisieren. Du kannst hierfür die Graph-Infrastruktur aus Übung 2 benutzen.

Die Kanten des Graphen sollen gewichtet werden. Dafür musst Du die Datenstruktur erweitern. Jede Kante soll symbolisch ein Gewicht bekommen, das anzeigt, wie „wahrscheinlich“ es ist, dass der Link benutzt wird. Benutze für eine Kante u, v das Gewicht $100 \cdot \text{indeg}(v) / \text{outdeg}(u)$, also Ein- und Ausgangsgrad der beiden Knoten (auf `int` gerundet).

2 Dijkstra (3 Punkte)

1. Der Dijkstra-Algorithmus funktioniert nicht mit negativen Kantengewichten. Erkläre wieso. (1 Punkt)
2. Implementiere den Dijkstra-Algorithmus. Du kannst ihn auf dem Wikipedia-Graphen testen, z.B. mit „Apple“ als Startknoten. (2 Punkte)

3 Topologische Sortierung (2 Punkte)

Implementiere topologische Sortierung für Graphen. Parse die Datei `frikadellen.txt` zu einem Testgraphen (gleiches Format wie `wikipedia.txt`).

Füge der Graph-Klasse eine Methode `tsort` hinzu, die die Knoten des Graphen topologisch sortiert in einer Liste zurück gibt. Du kannst die Ergebnisse Deiner Methode mit dem Output des UNIX-Tools `tsort` vergleichen, das Listen wie die in `frikadellen.txt` als Graph betrachtet und topologisch sortiert:

dhcp104-206: Michaela\$ tsort < frikadellen.txt

(Beachte hierbei, dass es mehrere mögliche topologische Sortierungen gibt.)

4 Cliques (Bonusaufgabe, 2 Punkte)

Implementiere den Basis-Algorithmus zur Extraktion von Cliques aus einem Graphen. Teste ihn auf dem Wikipedia-Graphen. Pseudocode für eine rekursive Funktion ist unten angegeben. (G = Graph, CS = aktuelle Cliquesmenge, CA = Kandidatenmenge, NOT = Visited, $CLIQUES$ = gefundene Cliques)

```
Procedure ExtractCliques( $G, CS, CA, NOT, CLIQUES$ )  
1 if  $CA = \emptyset$  AND  $NOT = \emptyset$  then  
2    $CLIQUES \leftarrow \{CS\} \cup CLIQUES$   
3 else  
4   if  $\exists a \in NOT$  s.t.  $\forall c \in CA : (a, c) \in E$  then  
5     return  
6 end  
7 if  $CA = \emptyset$  then  
8   return  
9 else  
10  forall  $ca \in CA$  do  
11    if  $ca \notin NOT$  then  
12       $CS' \leftarrow \{ca\} \cup CS$   
13       $CA \leftarrow CA \setminus \{ca\}$   
14       $NOT' \leftarrow \emptyset$   
15       $CA' \leftarrow \emptyset$   
16      forall  $a \in NOT$  do if  $(a, ca) \in E$  then  
17         $NOT' \leftarrow NOT' \cup \{CA\}$   
18      forall  $c \in CA$  do if  $(c, ca) \in E$  then  
19         $CA' \leftarrow CA' \cup \{ca\}$   
20         $NOT = NOT \cup \{ca\}$   
21        ExtractCliques( $G, CS', CA', NOT', CLIQUES$ )  
22    end  
23 end
```

Abgabe bis Freitag, 14.05.2010, 08:30 Uhr