Prof. Dr. Leif Kobbelt
Stefan Dollase, Ira Fesefeldt, Alexandra Heuschling, Gregor Kobsik

Übung 9

Tutoraufgabe 1 (Optimaler Suchbaum):

Gegeben sind folgende Knoten mit dazugehörigen Zugriffswahrscheinlichkeiten:

Knoten	I_0	N_1	I_1	N_2	I_2	N_3	I_3	N_4	I_4
Wert	$(-\infty,1)$	1	(1,2)	2	(2,3)	3	(3,4)	4	$(4,\infty)$
Wahrscheinlichkeiten	0.1	0.1	0.1	0.2	0.2	0.1	0.1	0.1	0.0

Konstruieren Sie einen optimalen Suchbaum wie folgt.

a) Füllen Sie untenstehende Tabellen für $W_{i,j}$ und $C_{i,j}$ nach dem Verfahren aus der Vorlesung aus. Geben Sie in $C_{i,j}$ ebenfalls **alle möglichen Wurzeln** des optimalen Suchbaums für $\{i, \ldots, j\}$ an.

$W_{i,j}$	0	1	2	3	4
1					
2	–				
3	–	–			
4	–	–	–		
5	–	–	–	–	

$C_{i,j}$ $(R_{i,j})$	0	1	2	3	4
1		( )	( )	( )	( )
2	–		( )	( )	( )
3	–	–		( )	( )
4	–	–	–		( )
5	–	–	–	–	

b) Geben Sie einen optimalen Suchbaum für die Knoten mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Knoten graphisch an.

c) Ist der optimale Suchbaum für die Knoten mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Knoten eindeutig? Geben Sie dazu eine kurze Begründung an.

Tutoraufgabe 2 (Union Find):

Führen Sie die folgenden Operationen beginnend mit einer anfangs leeren *Union-Find-Struktur* aus und geben Sie die entstehende Union-Find-Struktur nach jeder *MakeSet*, *Union* und *Find* Operation an. Nutzen Sie dabei die beiden Laufzeitverbesserungen: Höhenbalencierung und Pfadkompression. Dabei soll die Union-Operation bei **gleicher Höhe der Wurzeln immer die Wurzel des zweiten Parameters** als neue Wurzel wählen. Es ist nicht notwendig die Höhe der Bäume zu notieren.

1. MakeSet(1)
2. MakeSet(2)
3. MakeSet(3)
4. MakeSet(4)
5. MakeSet(5)
6. MakeSet(6)

7. MakeSet(7)
8. MakeSet(8)
9. MakeSet(9)
10. Union(1,2)
11. Union(3,4)
12. Union(3,1)
13. Union(5,6)
14. Union(7,8)
15. Union(7,9)
16. Union(9,5)
17. Union(9,2)
18. MakeSet(10)
19. Union(7,10)
20. Find(3)

Tutoraufgabe 3 (Prominenz suchen):

Sei ein gerichteter Graph $G = (V, E)$ als Adjazenzmatrix gegeben. Wir nennen einen Knoten $v \in V$ prominent, wenn von allen Knoten $v' \in V \setminus \{v\}$ eine Kante $(v', v) \in E$ nach v existiert, aber es von keinem Knoten $v' \in V$ eine Kante $(v, v') \notin E$ zurück gibt.
Geben Sie einen Algorithmus an, der in $O(|V|)$ Worst-case Laufzeit herausfindet, ob G einen prominenten Knoten besitzt. Begründen Sie die Korrektheit und die Laufzeit Ihres Algorithmus.

Aufgabe 4 (Optimaler Suchbaum): 7 + 2 + 1 = 10 Punkte

Gegeben sind folgende Knoten mit dazugehörigen Zugriffswahrscheinlichkeiten:

Knoten	I_0	N_1	I_1	N_2	I_2	N_3	I_3	N_4	I_4	N_5	I_5
Wert	$(-\infty,1)$	1	(1,2)	2	(2,3)	3	(3,4)	4	(4,5)	5	$(5,\infty)$
Wahrscheinlichkeiten	0.1	0.01	0.1	0.01	0.1	0.04	0.2	0.04	0.2	0.05	0.15

Konstruieren Sie einen optimalen Suchbaum wie folgt.

a) Füllen Sie untenstehende Tabellen für $W_{i,j}$ und $C_{i,j}$ nach dem Verfahren aus der Vorlesung aus. Geben Sie in $C_{i,j}$ ebenfalls **alle möglichen Wurzeln** des optimalen Suchbaums für $\{i, \ldots, j\}$ an.

$W_{i,j}$	0	1	2	3	4	5
1						
2	–					
3	–	–				
4	–	–	–			
5	–	–	–	–		
6	–	–	–	–	–	

$C_{i,j}$ ($R_{i,j}$)	0	1	2	3	4	5
1		( )	( )	( )	( )	( )
2	–		( )	( )	( )	( )
3	–	–		( )	( )	( )
4	–	–	–		( )	( )
5	–	–	–	–		( )
6	–	–	–	–	–	

b) Geben Sie einen optimalen Suchbaum für die Knoten mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Knoten graphisch an.

c) Ist der optimale Suchbaum für die Knoten mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Knoten eindeutig? Geben Sie dazu eine kurze Begründung an.

Aufgabe 5 (Union Find): 12 Punkte

Führen Sie die folgenden Operationen beginnend mit einer anfangs leeren *Union-Find-Struktur* aus und geben Sie die entstehende Union-Find-Struktur nach jeder *MakeSet*, *Union* und *Find* Operation an. Nutzen Sie dabei die beiden Laufzeitverbesserungen: Höhenbalencierung und Pfadkompression. Dabei soll die Union-Operation bei **gleicher Höhe der Wurzeln immer die Wurzel des zweiten Parameters** als neue Wurzel wählen. Es ist nicht notwendig die Höhe der Bäume zu notieren.

1. MakeSet(1)
2. MakeSet(2)
3. Union(1,2)
4. MakeSet(3)
5. Union(1,3)
6. MakeSet(4)
7. MakeSet(5)
8. Union(4,5)
9. Union(1,4)
10. MakeSet(6)
11. Union(3,6)
12. MakeSet(7)
13. MakeSet(8)
14. Union(7,8)
15. Union(2,7)
16. Find(7)

Aufgabe 6 (Graph Terminology): 2 + 2 + 2 + 2 + (4 * 0.5) = 10 Punkte

a) Sei V eine feste Knotenmenge mit Größe $|V| = n \in \mathbb{N}$. Wie viele Kantenmengen E gibt es, sodass (V, E) ein *gerichteter* Graph ist? Begründen Sie Ihre Antwort kurz.

b) Sei V eine feste Knotenmenge mit Größe $|V| = n \in \mathbb{N}$. Wie viele Kantenmengen E gibt es, sodass (V, E) ein *ungerichteter* Graph ist? Begründen Sie Ihre Antwort kurz.

c) Wie viele einfache Weg der Länge genau $k \in \{0, 1, \ldots, n-1\}$ hat ein vollständiger ungerichteter Graph mit $n \in \mathbb{N}$ Knoten? Begründen Sie Ihre Antwort kurz.

d) Ein einfacher Kreis $v_0 \ldots v_{k-1} v_0$ ist ein Kreis, für den $v_0 \ldots v_{k-1}$ einfach ist. Wie viele einfachen Kreise der Länge mindestens 3 hat ein vollständiger ungerichteter Graph mit $n \in \mathbb{N}$ Knoten? Begründen Sie Ihre Antwort kurz.

e) Sei $G = (V, E)$ ein gerichteter Graph. Wir definieren die Menge $E' = \{(i, j) \mid (j, i) \in E\}$. Betrachten Sie die Graphen $G^T = (V, E')$ und $\hat{G} = (V, \hat{E})$ mit $\hat{E} = E \cup E'$. Beweisen oder widerlegen Sie folgende Aussagen:

i) $\hat{G}$ ist symmetrisch.

ii) Falls $\hat{G}$ stark zusammenhängend ist, dann ist G oder G^T stark zusammenhängend.

iii) Falls G oder G^T stark zusammenhängend ist, dann ist auch $\hat{G}$ stark zusammenhängend.

iv) G ist schwach zusammenhängend genau dann, wenn G^T schwach zusammenhängend ist.

Hinweise:

- Die Länge eines Kreises $v_0 \ldots v_k$ ist k.
- Sie dürfen die Anzahlen auch mit $\sum$ und $\prod$ Termen angeben.

Aufgabe 7 (Zykel finden): 2 + 2 + 2 + 2 = 8 Punkte

Gegeben sei eine einfach verkettete Liste mit n Elementen, deren Länge Sie nicht kennen. Wir betrachten diese Liste im folgenden als gerichteten Graph.

a) Entwerfen Sie einen Algorithmus, mit dem sich testen lässt, ob der Graph einen Zykel enthält.

b) Zeigen Sie die Korrektheit Ihres Algorithmus.

c) Wie ist seine Laufzeit? Begründe Sie Ihre Antwort.

d) Ist dies auch in Zeit $O(n)$ möglich? Begründe Sie Ihre Antwort.