Übungsblatt 8

Datenstrukturen und Algorithmen (SS 2016)

Abgabe: Mittwoch, 15.06.2016, 23:59 Uhr — Besprechung: ab Montag, 20.06.2016

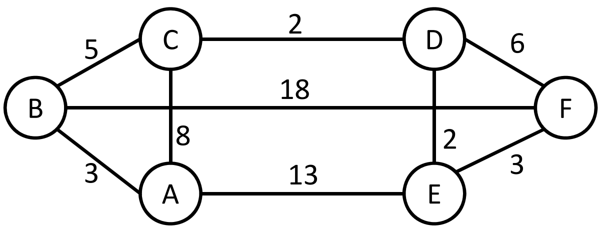
Bitte lösen Sie die Übungsaufgabe in **Gruppen von 3 Studenten** und wählen EINEN Studenten aus, welcher die Lösung im ILIAS als **PDF** als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt. Bitte erstellen Sie dazu ein **Titelblatt**, welches die Namen der Studenten, die Matrikelnummern, und die E-Mail-Adressen enthält.

Die Aufgaben mit Implementierung sind mit Impl gekennzeichnet. Das entsprechende Eclipse-Projekt kann im ILIAS heruntergeladen werden. Bitte beachten Sie die Hinweise zu den Implementierungsaufgaben, die im ILIAS verfügbar sind.[1]

Dieses Übungsblatt beinhaltet 5 Aufgaben mit einer Gesamtzahl von 30 + 10 Punkten (30 Punkte = 100%).

Aufgabe 1 Dijkstra-Algorithmus *[Punkte: 9]*

Gegeben ist der folgende gewichtete Graph $\mathcal{G}_0$



(a) *(6 Punkte)* Wenden Sie den *Dijkstra-Algorithmus* an, um den kürzesten Pfad vom Knoten A zu allen Knoten im Graphen $\mathcal{G}_0$ zu ermitteln. Vervollständigen Sie die folgende Tabelle durch Angabe der Kosten zum Erreichen der Knoten in jedem Schritt des Algorithmus.

Schritt	Kosten					
	A	B	C	D	E	F
Initialisierung						
1						
2						
3						
4						
5						

(b) *(3 Punkte)* Zeichen Sie den minimalen Spannbaum des Graphen $\mathcal{G}_0$

[1] `https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_fold_997779.html`

Aufgabe 2 Delaunay-Triangulierung *[Punkte: 11]*

In dieser Aufgabe sollen Sie eine gültige Triangulierung für eine gegebene Punktmenge finden. Hierfür werden die aus der Vorlesung bekannten Algorithmen *Plane-Sweep* und *Edge-Flip* verwendet.

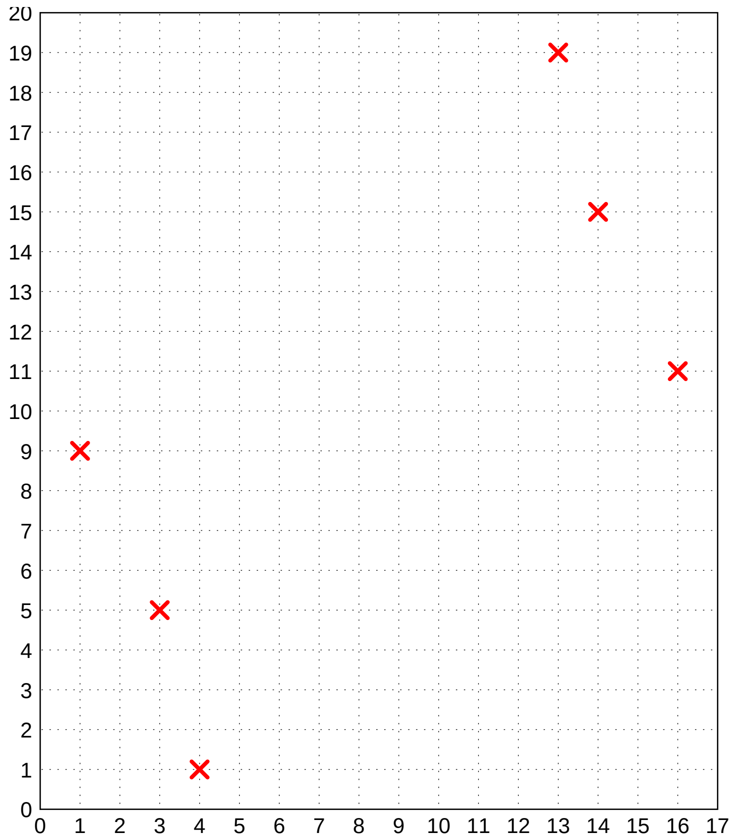


Abbildung 1: Für die dargestellten Punkte soll eine gültige Delaunay-Triangulierung erstellt werden.

(a) *(5 Punkte)* Führen Sie den Plane-Sweep-Algorithmus durch, um eine initiale Triangulierung für die Punkte in Abbildung 1 zu finden und reichen Sie die gezeichnete Triangulierung als Lösung ein.

(b) *(6 Punkte)* Führen Sie mit der initialen Triangulierung aus Aufgabenteil (a) den Edge-Flip-Algorithmus durch, bis das erzeugte Dreiecksnetz den Delaunay-Eigenschaften genügt.

- Markieren Sie in der initialen Triangulierung alle Kanten, die die lokale Delaunay-Eigenschaft verletzen.
- Zeichnen Sie danach alle Zwischenzustände auf, bei denen die lokale Delaunay-Eigenschaft vor einer Kantenvertauschung verletzt wird.
- Reichen Sie zudem den Endzustand des Dreiecksnetzes ein.

Falls Sie die vorige Teilaufgabe nicht bearbeitet haben, so wählen Sie bitte eine beliebige valide initiale Triangulierung.

Aufgabe 3 Impl Layoutalgorithmus für Binärbäume *[Punkte: 10]*

Auf Übungsblatt 3 wurde eine Datenstruktur für binäre Suchbäume implementiert. In dieser Aufgabe geht es nun darum, einen Binärbaum zu visualisieren. Hierfür sollen Sie die Positionen der Knoten bestimmen. Wir haben dazu die Klasse *BinaryTreeNode* um ein Positionsattribut erweitert, das aus einer x- und y-Koordinate besteht. Ihre Aufgabe ist es, diese Koordinaten nach folgendem Schema zu berechnen, damit Bäume wie in Abbildung 2 gezeigt entstehen.

Implementieren Sie in der Klasse *BinarySearchTree* die Methode *calculatePositions(...)*. Jedem Knoten soll als x-Position sein Index innerhalb einer *In-order*-Traversierung des Baums zugewiesen werden, startend mit Index 0 auf der linken Seite. Die y-Position eines Knotens ergibt sich durch die Ebene im Baum, wobei die Wurzel des Baums Ebene 0 trägt.

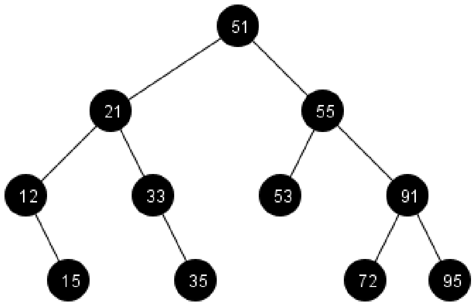
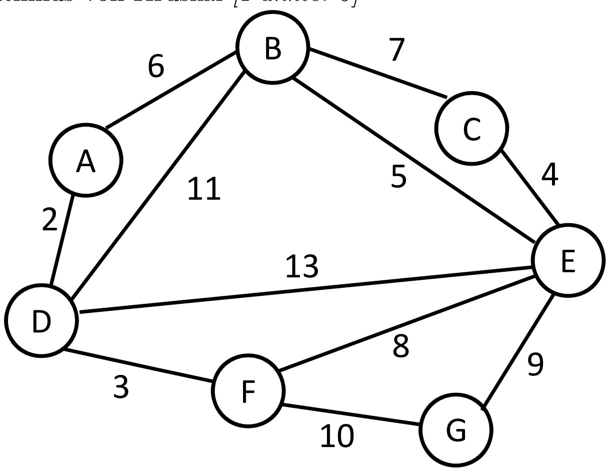


Abbildung 2: Beispielbaum, der mittels des Layoutalgorithmus erstellt wurde.

Für den in Abbildung 2 gezeigten Baum ergibt sich also beispielsweise für Knoten 12 die Position (x: 0, y: 2) und für Knoten 15 Position (x: 1, y: 3).

Wenn Sie alles richtig gemacht haben, sollten Sie beim Ausführen der Klasse *TreePanel* den Beispiel-Baum (Abbildung 2) sehen.

Aufgabe 4 (Bonus) Algorithmus von Kruskal *[Punkte: 5]*



Gegeben ist der gewichtete, ungerichtete Graph $\mathcal{G}_1$. Bestimmen Sie den minimalen Spannbaum mittels des Algorithmus von Kruskal (Foliensatz V14). Geben Sie dafür zunächst die sortierte Kantenliste an (Bsp: (AB,6)) und markieren sie mit einem + oder einem −, ob die Kante in dem minimalen Spannbaum enthalten ist. Zeichnen Sie auch den resultierenden Spannbaum.

Aufgabe 5 (Bonus) Literaturrecherche *[Punkte: 5]*

Recherchieren Sie im Internet nach der Originalpublikation des Dijkstra-Algorithmus von 1959, welcher ursprünglich nur den kürzesten Weg zwischen zwei Knoten bestimmt hat.

(a) *(2 Punkte)* Geben Sie die Anzahl der Schritte aus der Publikation an, welche Dijkstra zur Lösung des ersten Problems angibt. Geben Sie zusätzlich die Anzahl der Anmerkungen zur Lösung des zweiten Problems aus der Veröffentlichung an.

(b) *(1 Punkt)* Laden sie den BibTeX Eintrag für die Publikation von der Google Scholar Seite herunter und geben Sie diesen an.
Hinweis: Sie müssen möglicherweise die Einstellungen in Google Scholar anpassen.

(c) *(1 Punkt)* Laden sie den BibTeX Eintrag für die Publikation direkt von der entsprechenden Webseite des Herausgebers herunter und geben Sie diesen an.

(d) *(1 Punkt)* Vergleichen Sie die BibTeX Einträge von Google Scholar und der Webseite des Herausgebers. Was fällt Ihnen auf? Was schlussfolgern Sie daraus?