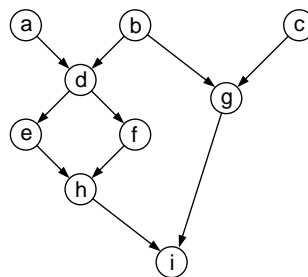


Übungsblatt 8 für “Diskrete und geometrische Algorithmen”

- 1.) Eine Frage zur Klassifikation der Kanten im Tiefensuchwald für die Tiefensuche in einem Graphen $G = (V, E)$, wobei $u.d$ den Entdeckungszeitpunkt und $u.f$ den Endzeitpunkt des Knotens $u \in V$ bezeichnet. Zeigen Sie, daß die Kante (u, v)
 - genau dann eine Baum- oder Vorwärtskante ist, wenn $u.d < v.d < v.f < u.f$ gilt,
 - genau dann eine Rückwärtskante ist, wenn $v.d \leq u.d < u.f \leq v.f$ gilt,
 - genau dann eine Querkante ist, wenn $v.d < v.f < u.d < u.f$ gilt.

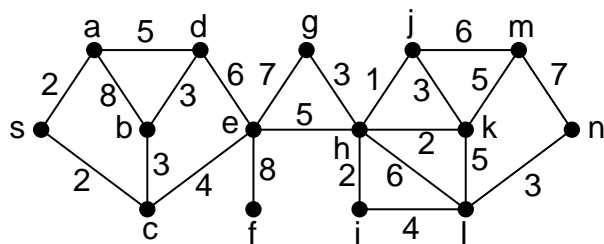
- 2.) Man erkläre die Funktionsweise des in der Vorlesung kennengelernten Algorithmus TOPOLOGICAL-SORT für das topologische Sortieren eines gerichteten azyklischen Graphens an Hand des nachfolgend angegebenen Graphen G :



- 3.) Geben Sie einen Algorithmus mit Laufzeit $\mathcal{O}(|V| + |E|)$ an, dessen Eingabe ein gerichteter azyklischer Graph $G = (V, E)$ und zwei Knoten s und t sind und der die Anzahl der (gerichteten) Wege (= einfache Pfade) von s nach t in G berechnet (Whg.: ein Weg ist eine Kantenfolge, wo alle Knoten entlang der Kantenfolge verschieden sind). Beispielsweise enthält der gerichtete azyklische Graph aus Bsp. (1) drei Wege vom Knoten b zum Knoten i , nämlich $b-d-e-h-i$, $b-d-f-h-i$ und $b-g-i$. Anmerkung: ihr Algorithmus soll die Pfade nur zählen, nicht ausgeben.

- 4.) Sei e eine Kante mit maximalem Gewicht auf einem Zyklus eines zusammenhängenden, kantenbewerteten Graphen $G = (V, E)$. Beweisen Sie, daß es einen minimalen Spannbaum von $G' := (V, E - \{e\})$ gibt, der gleichzeitig ein minimaler Spannbaum von G ist. Das heißt, es gibt einen minimalen Spannbaum von G , der e nicht enthält.

- 5.) Man erkläre die Funktionsweise der Algorithmen von Kruskal und Prim an Hand der Konstruktion eines maximalen Spannbaums (also eines spannenden Baums mit maximalem Gewicht) mittels Kruskal's Algorithmus (wobei selbstverständlich die Kanten absteigend nach ihren Gewichten sortiert wurden) und eines minimalen Spannbaums mittels Prim's Algorithmus für den Wurzelknoten (= Startknoten) s im folgenden kantenbewerteten Graphen G :



6.) Alternative Algorithmen zur Bestimmung minimaler Spannbäume. Nachfolgend sind die Pseudocodes von drei verschiedenen Algorithmen angegeben. Alle drei erhalten als Eingabe einen zusammenhängenden kantenbewerteten Graphen und geben eine Kantenmenge T zurück. Sie sollen für jeden der Algorithmen entweder beweisen, daß T in jedem Fall ein minimaler Spannbaum ist oder daß T kein minimaler Spannbaum sein muß.

```

MAYBE-MST-A( $G, w$ )
sortiere die Kanten in nichtsteigender Reihenfolge nach ihren Gewichten  $w$ 
 $T := E$ 
FOR jede Kante  $e$ , in der eben berechneten Reihenfolge betrachtet DO
  IF  $T - \{e\}$  ist ein zusammenhängender Graph THEN
     $T := T - \{e\}$ 
  END IF
END DO
RETURN  $T$ 

```

```

MAYBE-MST-B( $G, w$ )
 $T := \emptyset$ 
FOR jede Kante  $e$ , in einer beliebigen Reihenfolge betrachtet DO
  IF  $T \cup \{e\}$  enthält keine Zyklen THEN
     $T := T \cup \{e\}$ 
  END IF
END DO
RETURN  $T$ 

```

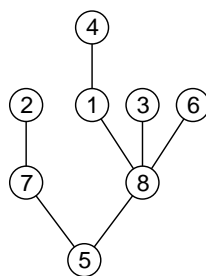
```

MAYBE-MST-C( $G, w$ )
 $T := \emptyset$ 
FOR jede Kante  $e$ , in einer beliebigen Reihenfolge betrachtet DO
   $T := T \cup \{e\}$ 
  IF  $T$  enthält einen Zyklus  $c$  THEN
    sei  $e'$  eine Kante von  $c$  mit maximalem Gewicht
     $T := T - \{e'\}$ 
  END IF
END DO
RETURN  $T$ 

```

7.) Dazu passend ein bißchen Kombinatorik. Betrachten wir den vollständigen Graphen $K_n = (V, E)$, mit $V = \{1, 2, \dots, n\}$ und $E = \{\{i, j\} : 1 \leq i < j \leq n\}$, also einen ungerichteten, einfachen (also keine Schlingen und keine Mehrfachkanten) Graphen, wo je zwei verschiedene Knoten durch eine Kante verbunden sind. Es gilt nun, daß K_n genau n^{n-2} verschiedene spannende Bäume besitzt. Um dies zu zeigen, betrachten wir markierte Wurzelbäume, also Bäume

mit Knotenmenge $\{1, 2, \dots, n\}$, wo ein Knoten als Wurzel ausgezeichnet ist und wo die Links-Rechts-Reihenfolge der Kinder eines Elternknotens irrelevant ist. Wenn wir zeigen können, daß es n^{n-1} verschiedene markierte Wurzelbäume mit n Knoten gibt, haben wir obige Behauptung bewiesen, da ja in einem spannenden Baum jeder der n Knoten als Wurzel ausgezeichnet werden kann und dann einen markierten Wurzelbaum darstellt. Die Abzählung von markierten Wurzelbäumen gelingt mittels sogenannten Prüfer-Codes, wo jedem markierten Wurzelbaum mit n Knoten eine Folge $(a_1, a_2, \dots, a_{n-1})$, mit $a_i \in \{1, 2, \dots, n\}$ für $1 \leq i \leq n-1$ zugeordnet wird. Um den Code zu generieren, entfernt man ausgehend vom Wurzelbaum T sukzessive Knoten und zwar im Schritt i das Blatt (= Knoten ohne Kinder) mit größtem label (unter allen Blättern) und merkt sich das **label des Elternknotens** als a_i . Beispielsweise ist der Prüfer-Code im nachfolgend angegebenen Wurzelbaum gegeben als $(8, 1, 8, 7, 5, 8, 5)$.



Um zu zeigen, daß eine Bijektion zwischen den markierten Wurzelbäumen mit n Knoten und den n^{n-1} Zahlenfolgen $\in \{1, 2, \dots, n\}^{n-1}$ vorliegt, muß man sich noch überlegen, wie man von einer gegebenen Zahlenfolge $(a_1, a_2, \dots, a_{n-1})$, mit $a_i \in \{1, 2, \dots, n\}$ für $1 \leq i \leq n-1$ den entsprechenden Wurzelbaum "zurückgewinnen" kann. Geben Sie einen Algorithmus (verbale Beschreibung oder Pseudocode genügt) an, wie dies gelingt.

Hinweis: Klarerweise entsprechen die Elemente, welche nicht in der Zahlenfolge (a_1, \dots, a_{n-1}) vorkommen, den labels der Blätter im Baum. Man überlege sich weiter, daß, wenn ein label das letzte Mal in der Zahlenfolge auftritt, es dann zu einem Blatt geworden ist.

- 8.) Man erkläre die Funktionsweise von Dijkstra's Algorithmus an Hand der Konstruktion eines sogenannten Entfernungsbauums (also des Vorgängerteilgraphs) vom Startknoten a zu allen übrigen Knoten im folgenden kantenbewerteten Graphen G :

