Ausgabedatum: 20.12.2012
Version: 2012-12-20 14:16

11. Übungsblatt zur Vorlesung Entwurf und Analyse von Algorithmen, WS 12/13

Abgabe: *Bis Donnerstag, 10.01.2013, 12:00 Uhr, Abgabenkasten im Treppenhaus 48-6.*

66. Aufgabe

Track ε 2
Track AI 2

Beschreiben Sie, wie Prims Algorithmus möglichst effizient implementiert werden kann, und bestimmen Sie anschließend die Worst-Case Laufzeit (in $\mathcal{O}$-Notation) Ihrer Implementierung (als Funktion der Anzahl Knoten und Kanten des verarbeiteten Graphen).

Hinweis: Eine möglichst effiziente Implementierung muss darauf achten, dass die *billigste kreuzende Kante* schnell bestimmt werden kann.

67. Aufgabe

Track ε 3
Track AI 3

Wir betrachten folgenden Algorithmus zur Berechnung eines minimalen Spannbaumes: Wie in Kruskals Algorithmus konstruieren wir den Baum, indem wir Kanten in einen Wald von Teil-Spannbäumen einfügen. Dabei gehen wir in Phasen vor, wobei wir in jeder Phase mehrere Kanten einfügen. Im Detail suchen wir in jeder Phase für jeden vorliegenden Teil-Spannbaum die kürzeste Kante, die ihn mit einem anderen Teil-Spannbaum des Waldes verbindet. Dann fügen wir all diese Kanten in den zu konstruierenden Baum ein (dieses Vorgehen ist als Boruvkas Algorithmus bekannt).

Zeigen Sie, dass Boruvkas Algorithmus einen minimalen Spannbaum konstruiert, wenn alle Kantengewichte paarweise verschieden sind. Beschreiben Sie, wie Boruvkas Algorithmus möglichst effizient implementiert werden kann und bestimmen Sie anschließend die Worst-Case Laufzeit (in $\mathcal{O}$-Notation) Ihrer Implementierung (als Funktion der Anzahl Knoten und Kanten des verarbeiteten Graphen).

68. Aufgabe

	a)	b)	c)
Track ε	[1]	[2]	[3]
Track AI	1	2	[3]

Erinnern Sie sich noch an die Enten aus Aufgabe 19? In dieser Aufgabe wollen wir untersuchen, wie wir die damals auf einer Modellierung mit Arrays basierenden Algorithmen mit seitdem gewonnen Erkenntnissen ausstechen können.

a) Modellieren Sie das Szenario mittels (Di)Graphen; berücksichtigen Sie dabei die weiteren Teilaufgaben!

Wie lösen Sie 19 a) in diesem Modell?

b) Entwerfen Sie einen Algorithmus im Modell von a), der das Problem aus 19 b) löst. Begründen Sie die Korrektheit des Algorithmus und analysieren Sie seine Laufzeit!

c) Entwerfen Sie einen Algorithmus im Modell von a), der das Problem aus 19 c) löst. Begründen Sie die Korrektheit des Algorithmus und analysieren Sie seine Laufzeit!

Hinweis: Das zugrundeliegende Problem ist in der Graphtheorie als *Graph Connectivity* bekannt.

69. Aufgabe

	a)	b)
Track ε	2	1
Track AI	[2]	[1]

Sei $G = (V, E, g)$ ein gewichteter Graph und $T = (V, E')$ ein Spannbaum von G. Wir bezeichnen die Kante

$$e^*(T) := \arg\max_{e \in E'} g(e)$$

als *Engstelle* von T und entsprechend

$$g^*(T) := \max_{e \in E'} g(e)$$

als *Engstellengewicht* von T. Unter allen Spannbäumen von G nennen wir die mit minimalen Engstellengewichten $g^*(\_)$ auch *engstellenoptimale Spannbäume*.

a) Sei G ein Graph mit paarweise verschiedenen Kantengewichten und T ein Spannbaum von G. Zeigen oder widerlegen Sie:

$$T \text{ minimaler Spannbaum} \implies T \text{ engstellenoptimaler Spannbaum} .$$

b) Sei T ein Spannbaum von G. Zeigen oder widerlegen Sie:

$$T \text{ engstellenoptimaler Spannbaum} \implies T \text{ minimaler Spannbaum} .$$

70. Aufgabe

Track ε [3]
Track AI [3]

Sei $G = (V, E, g)$ ein gewichteter Graph und seien $T_1 = (V, E_1)$ und $T_2 = (V, E_2)$ minimale Spannbäume von G. Wir bezeichnen mit

$$\text{count}(A, c) = |\{e \in A \mid g(e) = c\}|$$

die Anzahl Kanten eines bestimmten Gewichts c in einer Kantenmenge $A \subseteq E$.

Zeigen oder widerlegen Sie: Für alle $c \in \mathbb{R}$ gilt

$$\text{count}(E_1, c) = \text{count}(E_2, c) \ .$$

71. Aufgabe

Track ε 3
Track AI 3

Betrachten Sie folgenden Algorithmus `Sort`, der auf ein als global definiertes Feld `A` des Typs `ARRAY[1..n] OF CARDINAL` angewendet wird:

```
Sort(i,j : CARDINAL);
  VAR c,k : CARDINAL;
BEGIN
  IF i+1 < j THEN
    k := FLOOR((j-i+1)/3);
    Sort(i, j-k);
    Sort(i+k, j);
    Sort(i, j-k);
  ELSE
    IF A[i] > A[j] THEN
      c := A[i];
      A[i] := A[j];
      A[j] := c;
    END;
  END;
END;
```

Wie viele Vergleiche werden für die Abarbeitung des Aufrufs `Sort(1,n)` im schlechtesten Fall durchgeführt? Geben Sie die entsprechende Θ-Klasse an.

Beweisen oder widerlegen Sie außerdem, dass der Aufruf `Sort(1,n)` das Feld `A` sortiert.

72. Aufgabe

Track ε [3]
Track AI [3]

Geben Sie ein Verfahren an, um eine zufällige Permutation auf der Menge $\{1, 2, \dots, n\}$ zu erzeugen. Dabei sollen alle Permutationen gleich wahrscheinlich sein.

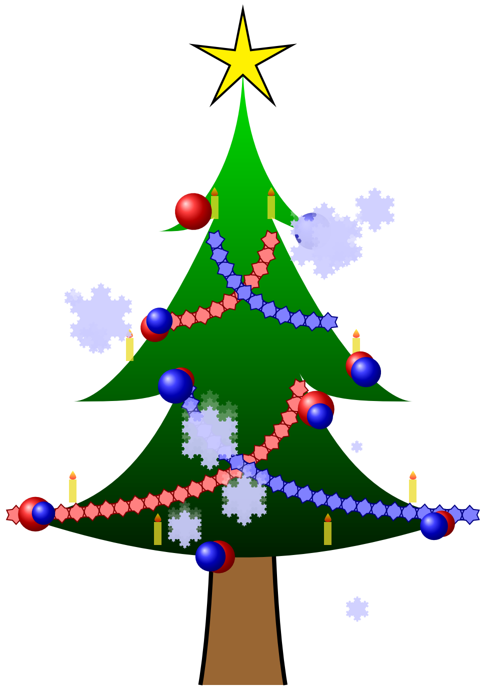
Beweisen Sie, dass Ihr Verfahren die gewünschten Eigenschaften hat, und analysieren Sie seine Laufzeit ($\mathcal{O}$-Klasse) sowie die Anzahl der benötigten Zufallsbits.

73. Aufgabe

Track ε [2]
Track AI [2]

Gegeben sei die Inversionentafel $\mathbf{b} = b_1, b_2, \dots, b_n$ der Permutation $\mathbf{a} = a_1, \dots, a_n$ auf der Menge $\{1, 2, \dots, n\}$. Beschreiben Sie einen Algorithmus, der aus der Eingabe $\mathbf{b}$ die entsprechende Permutation $\mathbf{a}$ rekonstruiert. Verdeutlichen Sie Ihr Verfahren, indem Sie seine Arbeitsweise anhand eines beliebigen Beispiels für den Fall $n = 10$ illustrieren.

Wir wünschen frohe und besinnliche Feiertage und einen guten Rutsch ins neue Jahr!



Graphik von Alain Matthes, siehe `http://tex.stackexchange.com/a/39211`