Übung 12 Abgabe 22.7.2014

Aufgabe 1 – Der RANSAC-Algorithmus für Kreise 18 Punkte

Gegeben ist das File "http://hci.iwr.uni-heidelberg.de/Staff/ukoethe/download/noisy-circles.txt", das zufällig verteilte 2D-Punkte enthält (ein Punkt pro Zeile), von denen einige deutlich Kreise markieren.

a) Benutzen Sie `gnuplot` oder ein anderes Programm Ihrer Wahl, um die Punkte zu zeichnen. In `gnuplot` sollte man durch den Befehl `"set size square"` ein quadratisches Ausgabefenster erzwingen (sonst werden die Kreise zu Ellipsen). Der entsprechende Befehl bei `matplotlib/pylab` lautet `"pylab.axes().set_aspect('equal')"`. *4 Punkte*

b) Implementieren Sie den RANSAC-Algorithmus für Kreise: *10 Punkte*

1. Wiederhole hinreichend oft:
   (A) Wähle zufällig drei Punkte und bestimme den Umkreis des dadurch definierten Dreiecks.
   (B) Zähle die Anzahl der "Inlier", d.h. die Anzahl der Punkte, deren Abstand von diesem Kreis höchstens *r* beträgt (für geeignet gewähltes *r*).
   (C) Wenn dieser Kreis mehr "Inlier" hat als der beste bisher bekannte, speichere den Kreis und die zugehörigen Inlier.
2. Falls weitere Kreise detektiert werden sollen: Entferne die Inlier des letzten Kreises aus der Liste und gehe zu 1.

Der Algorithmus soll als Funktion `circles = circleRANSAC(points, r, numberOfCircles)` implementiert werden, die im File `"circles.py"` abzugeben ist. `circles` ist ein Array, das jeden Kreis durch ein Tupel aus Mittelpunktskoordinaten und Radius repräsentiert.

c) Zeichnen Sie die detektierten Kreise ebenfalls in Ihr Diagramm ein und geben Sie das Diagramm als Bilddatei (z.B. PNG oder PDF) ab. In `gnuplot` verwendet man für Kreise den Befehl *4 Punkte*

```
>>> plot [0:2*pi] r*cos(t)+xcenter,r*sin(t)+ycenter
```

Die entsprechenden Befehle für `matplotlib/pylab` lauten:

```
>>> c = pylab.Circle((x,y), radius, fill=False)  # Kreis erzeugen
>>> pylab.gca().add_artist(c)                    # Kreis einfügen
>>> pylab.axis()                                 # Kreis anzeigen
```

Bonusaufgabe 2 – Indirektes Sortieren 8 Punkte

Wir haben in der Vorlesung gezeigt, dass man ein Array in linearer Zeit sortieren kann, wenn die Permutation der Arrayelemente bekannt ist (siehe Kapitel "Sortieren in linearer Zeit"). Mit einem geeigneten Funktor `PermutationSortFunktor` kann die universelle Funktion `array.sort()` auch dazu verwendet werden, eine solche Permutation zu bestimmen. Dies ist z.B. notwendig, wenn man nur lesenden Zugriff auf das zu sortierende Array hat, so dass `read_only_array.sort()` nicht erlaubt ist. Eine sortierte Ausgabe von `read_only_array` kann stattdessen durch indirekten Zugriff mittels der Permutation erfolgen:

```
read_only_array = ...                       # das zu sortierende Array
permutation = range(len(read_only_array))   # Init der Permutation
f = PermutationSortFunktor(...)             # Init Ihrer Funktor-Klasse
permutation.sort(f)                         # Bestimmen der Permutation
for k in xrange(len(read_only_array)):
    print read_only_array[permutation[k]]   # sortierte Ausgabe
```

Das Array `permutation` enthält also am Ende die Indizes von 0 bis `len(read_only_array)-1` in der richtigen Reihenfolge, so dass `read_only_array[permutation[k]]` gerade das k-te Element des *sortierten* Arrays ist. Implementieren Sie die Klasse `PermutationSortFunktor` (also insbesondere den

Konstruktor und die `__call__()`-Methode für den erforderlichen drei-wertigen Vergleich) und vervollständigen Sie den gegebenen Code. Implementieren Sie außerdem die Funktion `sortByIndexArray(read_only_array, permutation)` (siehe Wiki), die mit Hilfe der Permutation in linearer Zeit eine sortierte Kopie des `read_only_array` erzeugt und benutzen Sie diese Funktion, um geeignete Unit-Tests zu implementieren. Geben Sie Ihre Lösung im File `index_sort.py` ab.

Hinweise: (i) Bezüglich der generischen Sortierung vergleichen Sie Aufgabe 1 in Übung 9. (ii) Der Funktor `PermutationSortFunktor` muss offensichtlich auf die Daten in `read_only_array` zugreifen. Dies geschickt zu realisieren ist der eigentliche Clou der Aufgabe.

Bonusaufgabe 3 – Klausurvorbereitung 20 Punkte

Einige Aufgaben in der Klausur werden den folgenden Aufgaben ähneln.

a) Sortieren sie folgendes Array mittels Selection Sort: [5,3,6,2,1,4]. Geben sie dabei alle Zwischenschritte an. Was bedeutet "stabiles Sortieren", und gilt dies für den Selection Sort-Algorithmus? *3 Punkte*

b) Wodurch unterscheidet sich ein binärer Suchbaum von einem beliebigen Baum (mindestens zwei Unterschiede)? Welcher binäre Suchbaum ergibt sich, wenn man die folgende Liste in einen leeren Baum einfügt: [5,9,1,2,7,8,6,3,4]? *3 Punkte*

c) Geben sie für die folgenden Laufzeiten die Komplexitätsklasse mittels Master-Theorem an (in der Klausur wäre das Mastertheorem in der Aufgabe angegeben, damit Sie es nicht auswendig lernen müssen): *4 Punkte*

I. $T(n) = 3\,T\left(\frac{n}{2}\right) + n^2$

II. $T(n) = 16\,T\left(\frac{n}{4}\right)$

III. $T(n) = 3\,T\left(\frac{n}{3}\right) + \frac{n}{2}$

d) Ordnen sie die folgenden Ausdrücke nach ihrer Laufzeitkomplexität und zeigen Sie im Grenzfall $n \to \infty$, dass Ihre Anordnung korrekt ist (die dafür nützlichen Differentiationsregeln werden in der Klausur angegeben, falls eine solche Aufgabe drankommt): *5 Punkte*

I. $\sqrt{5n}$

II. $\log(n^3 + n)$

III. $n^2 + 4n + 8$

IV. $2^{\frac{n}{2}}$

e) Gegeben sind die folgenden Tupel: (19, 7, 13), (10, 19, 10), (19, 11, 15), (3, 18, 11), (13, 7, 19), (14, 8, 14), (19, 14, 1), (10, 18, 3), (6, 11, 1), (3, 15, 15), (2, 10, 14) sowie die drei Hashfunktionen: *5 Punkte*

I. $\left(\sum_{k=0}^{2} a_k\, x_k\right) \bmod 11 \quad \text{mit} \quad a = [6,6,2]$

II. $\left(\sum_{k=0}^{2} a_k\, x_k\right) \bmod 13 \quad \text{mit} \quad a = [5,1,8]$

III. $\left(\sum_{k=0}^{2} 2^k\, x_k\right) \bmod 11$

wobei *x* das Tupel ist, dessen Hash man gerade berechnet, und x_k sein k-tes Element. Was ist eine minimale perfekte Hashfunktion? Welche der drei Funktionen ist eine solche, und warum gilt dies für die anderen Funktionen nicht? Wie lautet die Hashtabelle, die man unter Benutzung der geeigneten Funktion erhält, wenn man alle gegebenen Tupel einfügt?