Übungen zu Computergrundlagen WS 2014/2015

Übungsblatt 4: Python

05.11.2014

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
  - **Freitag, 14.11.2014, 10:00**
- Schickt die Lösungen bitte per Email an Euren Tutor.

Aufgabe 4.1: Sudoku (5 Punkte)

	7	3					2	
				1		9		
9					8	5		
	4		2					
			3		7			
5				4				6
							3	
1								

Abbildung 1: Beispiel-Sudoku

In dieser Aufgabe geht es darum, die Schritte, die vor der eigentlichen Programmierarbeit stattfinden, zu verinnerlichen. Dass heißt, es ist nicht nötig, am Ende ein funktionierendes Programm abzugeben.

Stelle dir vor du müsstest ein Programm schreiben, dass ein gegebenes Sudoku-Puzzle löst. Wer Sudoku noch nicht kennt: `http://en.wikipedia.org/wiki/Sudoku`.

- **4.1.1** Führe eine gründliche Problemanalyse durch. Was soll das Progamm leisten? D.h., was bedeutet es genau, ein Sudoku zu lösen? Was sind die Nebenbedingungen? Was sind die Eingabedaten? (2 Punkte)
- **4.1.2** Diskutiere, mit welcher Methode Du das Problem *algorithmisch* lösen würdest. Welche Teilprobleme lassen sich identifizieren? Welche Datenstrukuren bieten sich an und warum? (3 Punkte)

Hinweise: Überlege Dir, was die Bedingungen an eine gültige Lösung eines Sudokus sind und wie Du diese algorithmisch prüfst. Überlege Dir dann, wie Du mögliche Lösungen erzeugst.

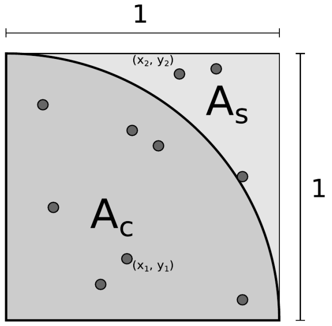

Abbildung 2: Eine Monte Carlo-basierte Methode zur Berechnung der Kreiszahl π.

Aufgabe 4.2: πthon (5 Punkte)

In dieser Aufgabe geht es um die (näherungsweise) Berechnung der Kreiszahl π mithilfe von Computerprogrammen.

Eine einfache Methode, um eine Abschätzung für die Kreiszahl π zu erhalten, ist die in Abbildung 2 skizziert. Die Idee dabei ist es, zufällig N_s Punkte (x_i, y_i) aus einem Einheitsquadrat (Quadrat mit Kantenlänge 1) zu ziehen ($0 \leq x_i \leq 1$, $0 \leq y_i \leq 1$). N_c sei die Anzahl der Punkte davon, die in einem Viertel des Einheitskreises (Kreis mit Radius 1) liegen (also bei denen $x^2 + y^2 < 1$ ist). Da wir wissen, daß die Fläche des Einheitsquadrats $A_s = 1$ ist, und die Fläche des Einheitskreises $A_c = \pi$, sollte also gelten

$$\frac{N_c}{N} \approx \frac{\frac{1}{4}A_c}{A_s} = \frac{1}{4}\pi \Rightarrow \pi \approx 4\frac{N_c}{N}$$

Wir nähern damit die Fläche des Kreissegments an. Dabei gilt: je größer N, desto genauer ist die Abschätzung. Wir können π also durch folgende Formel annähern:

$$\pi \approx 4\frac{1}{N}\sum_{i=1}^{N} \chi_{x^2+y^2<1}(x_i, y_i)$$

wobei

$$\chi_{x^2+y^2<1}(x, y) = \begin{cases} 1, & \text{falls } x^2 + y^2 < 1 \\ 0, & \text{sonst} \end{cases}$$

- **4.2.1** Führe hier ebenfalls eine Problemanalyse durch und formuliere einen Algorithmus, basierend auf den obigen Überlegungen. Welche Eingabedaten brauchst Du? (2 Punkte).
- **4.2.2** Implementiere die hier dargestellte Methode zur näherungsweisen Bestimmung der Kreiszahl π in Python. (3 Punkte)

Hinweise:

- Um in Python Zufallszahlen zu erzeugen, braucht ihr das Modul `random`. Dessen Funktion `random()` erzeugt gleichverteilte Zufallszahlen zwischen 0 und 1, die Ihr wie folgt erhaltet:

```
from random import random
zufall = random()
```