

7. Aufgabenblatt

Der Abgabetermin für Ihre Lösungen ist Montag, der **26. Juni 2017** um 23⁵⁵ Uhr. Die Bewertung erfolgt auf der Grundlage Ihres `git`-Repositories zu diesem Zeitpunkt. Stellen Sie sicher, dass Sie bis zu diesem Zeitpunkt alle Änderungen in Ihr Repository übertragen haben und vergessen Sie nicht das `makefile`!

Aufgabe 17

Entwerfen Sie eine Klasse, die in der Form eines linearen Kongruenzgenerators

$$x_{i+1} = (ax_i + c) \bmod m \quad (1)$$

Zufallszahlen erzeugt.

- ▶ Erzeugen Sie mit den Parametern $a = 4$, $c = 0$, $m = 200$ und $x_0 = 1$ zufällige Werte. Was fällt auf, wenn Sie sich mehr als zehn Werte anzeigen lassen?
- ▶ Der Zufallsgenerator RANDU von IBM verwendet die Parameter $a = 65539$, $c = 0$, $m = 2^{31}$ und $x_0 = 1$. Lassen Sie sich ebenfalls die ersten Werte anzeigen.
- ▶ Führen Sie mit den ersten 30 000 Zahlen den Spektraltest in drei Dimensionen durch. Schreiben Sie hierfür je drei Zahlen pro Zeile in eine Datei und plotten Sie die Daten mittels `gnuplot` (Befehl: `splot`).
- ▶ Untersuchen Sie, ob die Punkte zufällig im Raum verteilt erscheinen und beschreiben Sie Ihre Beobachtung in ISIS.
- ▶ Lagern Sie Ihre Klasse zur späteren Wiederverwendung in eine separate `cpp`-Datei aus und erstellen Sie natürlich die passende Headerdatei.

(5 Punkte)

Aufgabe 18

Schreiben Sie eine Klasse für einen Zufallszahlengenerator, der auf der Kombination von Xorshift-Methode und multiplikativem linearen Kongruenzgenerator (MLCG) gemäß

$$\text{Ausgabe} = \text{MLCG}(\text{Xorshift}) \quad (2)$$

basiert. Die genaue Vorschrift für die Xorshift-Methode lautet:

$$\begin{aligned} x &\leftarrow x \wedge (x \gg a_1) \\ x &\leftarrow x \wedge (x \ll a_2) \\ x &\leftarrow x \wedge (x \gg a_3) \end{aligned}$$

mit $a_1 = 21$, $a_2 = 35$ und $a_3 = 4$.

Die Vorschrift für den MLCG entspricht mit $a = 2685821657736338717$, $c = 0$ und $m = 2^{64} - 1$ Formel 1 aus Aufgabe 17. Die Parameter a und m sind größer als der maximale Integerwert $2^{32} - 1$. Achten Sie daher darauf, den richtigen Datentyp zu verwenden und das Ganzzahlliteral entsprechend anzugeben.

- ▶ Im Konstruktor der Klasse soll ein Wert $v = 4101842887655102017$ gegen einen vom Benutzer einzugebenden Startwert x_0 mit XOR verknüpft und anschließend die Vorschrift (2) ausgeführt werden.
- ▶ Implementieren Sie eine Methode, die Fließkommazahlen in einem beliebigen, vom Benutzer zu bestimmenden Intervall $[\alpha, \beta]$ ausgibt.
- ▶ Lagern Sie Ihre Klasse zur späteren Wiederverwendung in eine separate `cpp`-Datei aus und erstellen Sie die passende Headerdatei.
- ▶ Führen Sie auch für diesen Zufallszahlengenerator den Spektraltest in zwei und drei Dimensionen durch. Was zeigt der Vergleich mit RANDU?

(6 Punkte)

Aufgabe 19

Schreiben Sie ein Programm, das mit der Verwerfungsmethode 100 000 Zufallszahlen vom Typ `double` entsprechend einer Gaußverteilung erzeugt.

- ▶ Speichern Sie die Zufallszahlen in einer Datei (`gauss.dat`).
- ▶ Schreiben Sie eine Funktion `histogramm`, die den Wertebereich in n gleich große Intervalle unterteilt und ein Histogramm erstellt.
- ▶ Speichern Sie die Histogrammdatei in einer Datei (`histogramm.dat`) und erstellen Sie mittels Gnuplot eine graphische Darstellung der Verteilung.

(6 Punkte)

Hinweise

- ▶ Sämtliche Parameter zur Erzeugung der Zufallszahlen (a , c , m und x_i für den MLCG und die a_n für die Xorshift-Methode) sind private Variablen, die vom Konstruktor initialisiert werden sollten. Implementieren Sie geeignete Methoden, um diese Werte zu setzen und auszugeben.
- ▶ Es bietet sich an, dem Konstruktor den Anfangswert x_0 für die Zufallszahlensequenz (*seed*) als Parameter zu übergeben.