Algorithmus zur Bestimmung einer zufälligen Permutation

Praktikum "Visuelle Wahrnehmung"; Pierre Bayerl & Heiko Neumann (WS 2003/2004)
Abteilung Neuroinformatik, Universität Ulm

Algorithmus "random_shuffle" (aus STL: .../bits/stl_algo.h) zum Erzeugen einer zufälligen Permutation des Arrays A[first..last] (vergleiche [1, 2]):

```
PROCEDURE random_shuffle(A, first, last)
  IF (first == last) RETURN;
  FOR i = first TO last
    SWAP( A[i],
          A[first + random_number(i - first + 1] );
```

`random_number(K)` erzeugt eine Zufallszahl zwischen 0 und $K-1$.

OBdA: Annahme initial $A[i] = i$ und first$=1$; A_N bezeichne ein Array der Länge N.

Zu zeigen: $P(A_N[i] = x) = \frac{1}{N}$ $\forall N$ und $1 \leq i, x \leq N$.

• Induktionsanfang:

$N = 1$: $P(A_1[1] = 1) = 1$ (trivial).

• Induktionsschritt:

$N \to N + 1$:

- Induktionsvorraussetzung $P(A_N[i] = x) = \frac{1}{N}$ $\forall 1 \leq i, x \leq N$.
- Zu zeigen: $P(A_{N+1}[i] = x) = \frac{1}{N+1}$ $\forall 1 \leq i, x \leq N + 1$.

Fall 1: $i, x < N + 1$:

$$\begin{aligned} P(A_{N+1}[i] = x) &= P(A_N[i] = x)P(random\_number(N+1) \neq i) \\ &= \frac{1}{N}\frac{N}{N+1} = \frac{1}{N+1} \end{aligned}$$

Fall 2: $i, x = N + 1$:

$$P(A_{N+1}[N+1] = N+1) = P(random\_number(N+1) = N+1) = \frac{1}{N+1}$$

Fall 3: $i = N + 1, x < N + 1$:

$$\begin{aligned} P(A_{N+1}[N+1] = x) &= \sum_{j=1}^{N} P(A_N[j] = x)P(random\_number(N+1) = j) \\ &= \sum_{j=1}^{N} \frac{1}{N}\frac{1}{N+1} = \frac{N}{N(N+1)} = \frac{1}{N+1} \end{aligned}$$

Fall 4: $i < N + 1, x = N + 1$:

$$P(A_{N+1}[i] = N+1) = P(random\_number(N+1) = i) = \frac{1}{N+1}$$

$\blacksquare$

Literatur:

[1] Standard Template Library Programmer's Guide (Dec. 2003). http://www.sgi.com/tech/stl

[2] D.E. Knuth. The Art in Computer Programming. Vol 2: Seminumerical Algorithms. Addison-Wesley, 2nd. Ed., 1981.