1. Das Teilchen-Fächer-Modell

Wir haben n Fächer und k Teilchen. Wir wollen die Möglichkeiten zählen, die Teilchen auf die Fächern zu verteilen.

a) Die Teilchen seien unterscheidbar (nummeriert von 1 bis k)

Mehrfachbelegung erlaubt

keine Mehrfachbelegung

b) Die Teilchen seien nicht unterscheidbar (weiße Kugeln)

Mehrfachbelegung erlaubt

keine Mehrfachbelegung

2. In einem Teig sind 10 Rosinen. Aus dem Teig werden 10 Semmeln gebacken und eine Semmel zufällig ausgewählt. Wie groß ist die Wahrscheinlichkeit, dass diese genau 2 Rosinen enthält ?

3. Auf 10 Fächer werden zufällig 4 Kugeln verteilt. Wie groß ist die Wahrscheinlichkeit, dass es ein Fach gibt, in dem mehrere Kugeln liegen ?

4. Gegeben sind die Permutationen:

$$\pi : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 5 & 1 \end{pmatrix} \qquad \gamma \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 5 & 1 & 3 \end{pmatrix}$$

Man berechne:

$$\pi \circ \gamma \,,\, \gamma \circ \pi \,,\, \pi^{-1} \text{ und } \gamma^{-1}$$

5. Es werden 10 Briefe und 10 Briefumschläge mit den Adressen geschrieben. Die Briefe werden zufällig in die Briefumschläge gesteckt. Wie groß ist die Wahrscheinlichkeit, dass mindestens ein Brief beim richtigen Empfänger ankommt.

6. Permutationen in lexikographischer Reihenfolge

Generieren Sie für eine natürliche Zahl n alle Permutationen der Menge $\{1, 2, \ldots, n\}$ in lexikographischer Reihenfolge.
Eingabe: In der Datei *genperm.in* befindet sich eine natürliche Zahl n ($1 \leq n \leq 10$).

Ausgabe: Schreiben Sie in die Datei *genperm.out* alle Permutationen der Menge $\{1, 2, \ldots, n\}$ in lexikographischer Reihenfolge, und zwar eine Permutation pro Zeile.

genperm.in	genperm.out
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

7. Ranking einer Permutation in lexikographischer Reihenfolge

Wenn eine Ordnung über die Menge aller n-Permutationen (die Permutationen mit n Elementen) gegeben ist, dann ist der Rang einer Permutation gleichbedeutend mit deren Stelle in der geordneten Liste aller n-Permutationen. Wenn die Rangordnung mit 0 anfängt, dann können wir jeder Permutation einen Rang zwischen 0 und $n!$-1 zuweisen. Wir suchen den Rang einer gegebenen n-Permutation bzgl. der lexikographischen Ordnung und nehmen an, dass er in den Typ *long* passt. Beispiel:

perm.in	rank.out
4 4 3 1 2	22
7 7 5 6 3 1 2 4	4908

8. Unranking einer Permutation in lexikographischer Reihenfolge

Das umgekehrte Problem. Wenn eine Ordnung über der Menge aller n-Permutationen gegeben ist, dann kennzeichnet die Stelle einer Permutation in der geordneten Liste

aller n-Permutationen den Rang dieser Permutation. Wenn die Rangordnung mit 0 anfängt, können wir jeder Permutation einen Rang zwischen 0 und $n!$-1 zuweisen. Wir wollen die n-Permutation bestimmen, für die n und ihr Rang gegeben sind. n und der Rang werden in Variablen des Typs *long* gespeichert. Beispiel:

rank.in	perm.out
4 22	4 3 1 2
7 4908	7 5 6 3 1 2 4

Literatur

1. Albrecht Beutelspacher, Marc-Alexander Zschiegner, *Diskrete Mathematik für Einsteiger. Mit Anwendungen in Technik und Informatik*, 3. Auflage, Vieweg Verlag, 2007.

2. Doina Logofătu, *Algorithmen und Problemlösungen mit C++*, Vieweg+Tebuner Verlag, 2010.