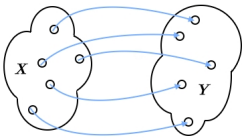
Einführung

Unter einer Permutation versteht man in der Mathematik die Veränderung der Reihenfolge der Elemente einer Menge. Zum Beispiel kann die Permutation eines bestimmten strings bei der Suche nach einem Passwort, wessen Buchstaben zwar bekannt sind nicht aber deren Reihenfolge, ganz nützlich sein.

Eine Permutation ist mathematisch gesehen eine bijektive Abbildung einer endlichen Menge mit n Elementen auf sich selbst $f : X \to Y$
. Es handelt sich daher einfach um eine Neuanaordnung der Elemente. Als solche ist sie eineindeutig und es existiert somit eine Umkehrfunktion. Das bedeutet nichts anderes als das sie aus der Zielmenge wieder eindeutig auf die Bildmenge schließen können.



Permutationen spielen nicht nur in der reinen Kombinatorik eine wichtige Rolle sondern finden auch in der Spieleentwicklung Verwendung. Darunter zum Beipiel bei der Matrizenberechnung von geometrischen Figuren.

Permutation ohne Wiederholung

Bei der Permutation ohne Wiederholung müssen folgende Voraussetzungen erfüllt sein:

- Alle (N) Elemente der Ausgangsmenge unterscheiden sich voneinander.
- Es müssen alle (N) Elemente ausgewählt werden.
- Ein Element kann nicht mehrmals ausgewählt werden.

Die Berechnung erfolgt über die Fakultätsbildung $P(n) = n!$
, wobei n stellvetretend für die Anzahl der Elemente steht. Es gilt $n \in \mathbb{N}_0$
. Falls ihnen der Begriff der Fakultät nicht geläufig ist, so können sie anhand der nachfolgenden Formel ersehen wie diese gebildet wird.

$$n! = \prod_{k=1}^{n} k = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Nehmen wir als Beispiel die folgende Situation an. Ein Einbrecher benötigt zum Öffnen einer durch ein Kombinationsschloß verriegelten Türe einen Code. Nach gründlicher Spurensuche findet er heraus, das sich auf dem Schloss nur auf vier Tasten Fingerabdrücke befinden. Nämlich auf Taste **3**, **4**, **5** und **8**. Außerdem weiß er das beim Codeschloß mit der Seriennummer 25F7-ST66 ein 4-stelliger Zugangscode in der richtigen Reihenfolge eingegeben werden muss. Die Frage lautet nun, wieviele Kombinationen muss unser Einbrecher durchgehen um das Schloß zu knacken und welche sind das?

Wie bereits oben erläutert errechnet sich die Anzahl der möglichen Kombinationen aus der Fakultät. Somit ergeben sich für vier unterschiedliche Elemente aus der Menge P insgesamt 24 Permutationen. Nachfolgend sind diese vollständig gelistet: 3458, 3485, 3548, 3584, 3845, 3854, 4358, 4385, 4538, 4583, 4835, 4853, 5348, 5384, 5438, 5483, 5834, 5843, 8345, 8354, 8435, 8453, 8534, 8543.

Nun ist das Ganze für vier Zahlen (Elemente) noch sehr überschaubar und die 24 verschiedenen Permutationen lassen sich problemlos in kurzer Zeit durchdenken. Doch aufgrund des Wachstumsverhaltens von n! steigt die Anzahl an möglichen Permutationen ab 5 Elementen sehr stark an und erreicht bereits für n=69 eine Zahl mit unglaublichen **99** Dezimalstellen! Deshalb wäre es gut ein Programm zu haben welches uns die Aufgabe der Berechnung abnimmt. Die C++ Standard Bibliothek mit ihren mächtigen Algorithmen stellt uns hierfür die Funktionen
prev_permutation
und
next_permutation
zur Verfügung. Diese sind in der STD Header-Datei algorithm versammelt. Der Algorithmus prev_permutation generiert für den Bereich [first, last] alle vorhergehenden Permutation der Elemente, während next_permutation dies für alle nachfolgenden macht.

Geordnete Elemente

Beachten sie das der STL Algorithmus davon ausgeht dass alle Elemente lexikographisch geordnet vorliegen! Existiert eine vor-/nachfolgende Permutation wird
true
zurückgegeben. Das untere Programm erzeugt für die Buchstaben A, B und C insgesamt sechs Permutationen.

[code xml:lang="cpp"]// Calculate Permutations - using STL Algorithms #include #include int main() { char f[] = { 'A', 'B', 'C' }, *const fn = f + sizeof(f) / sizeof(*f); unsigned int i = 0; do { std::cout