

Aufgabenblatt 4

Algorithmen und Datenstrukturen, WS 2015/16

Prof. Dr. Ulrich Hedtstück, HTWG Konstanz

Programmieren Sie einen Algorithmus zum Erzeugen aller Permutationen der Zahlen $1, \dots, n$, wobei n eine beliebige natürliche Zahl ist und die Zahlen $1, \dots, n$ in einem Array `int [] p` gegeben sind.

	1234	Beispiel ein Schritt im
	1243	Algorithmus für $n = 10$
	1324	
	1342	...
	1423	8 5 6 3 9 7 4 2 1
	1432	↓
Beispiel $n = 4$:	2134	8 5 6 4 1 2 3 7 9
	2143	...
	2314	
	2341	
	2413	
	2431	
	...	

Die Reihenfolge der Permutationen soll gemäß der durch $1 < 2 < \dots < n$ definierten *lexikographischen Ordnung* erfolgen, d.h. $1\ 2\ \dots\ (n-2)\ (n-1)\ n$ ist die erste Permutation, $1\ 2\ \dots\ (n-2)\ n\ (n-1)$ ist die zweite, und $n\ (n-1)\ (n-2)\ \dots\ 2\ 1$ ist die letzte Permutation in dieser Reihenfolge.

Der Algorithmus soll iterativ nach dem folgenden Prinzip programmiert werden:

- Schritt 1: Bestimme das größte i , $0 \leq i \leq n-1$, mit $p[i] < p[i+1]$.
- Schritt 2: Bestimme das kleinste $p[j]$ mit $i+1 \leq j \leq n-1$,
so daß $p[i] < p[j]$.
- Schritt 3: Vertausche $p[i]$ und $p[j]$.
- Schritt 4: Sortiere $p[i+1] \dots p[n-1]$ in aufsteigender Reihenfolge.

Da die Zahlen $p[i+1] \dots p[n-1]$ nach Schritt 3 in absteigender Reihenfolge dastehen, kann das Sortieren in Schritt 4 dadurch realisiert werden, dass man $p[i+1]$ mit $p[n-1]$ vertauscht, $p[i+2]$ mit $p[n-2]$, usw.

Testen Sie Ihr Programm dadurch, dass Sie für kleine n nach jedem Durchgang der vier Schritte das Ergebnis auf dem Bildschirm ausdrucken.

Ermitteln Sie den größten Wert für n , der von Ihrem Rechner noch in sinnvoller Zeit bearbeitet werden kann.

Das Programm soll in den Übungen vorgeführt und erklärt werden, spätestens am Freitag, den 04.12.2015.