

Permutationen

Übungen

Aufgabe 1

Notiere die Permutation p in vereinfachter Zykendarstellung.

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 3 & 9 & 8 & 2 & 11 & 12 & 5 & 4 & 1 & 10 & 7 & 6 \end{pmatrix}$$

Aufgabe 1

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 3 & 9 & 8 & 2 & 11 & 12 & 5 & 4 & 1 & 10 & 7 & 6 \end{pmatrix}$$

$$p = (1 \ 3 \ 8 \ 4 \ 2 \ 9) (5 \ 11 \ 7) (6 \ 12)$$

Aufgabe 2

Notiere die Permutation p in vereinfachter Zyklendarstellung.

$$p = \begin{pmatrix} 8 & 7 & 4 & 10 & 2 & 6 & 9 & 5 & 3 & 1 & 11 \\ 9 & 7 & 2 & 3 & 4 & 5 & 8 & 1 & 11 & 6 & 10 \end{pmatrix}$$

Aufgabe 2

$$p = \begin{pmatrix} 8 & 7 & 4 & 10 & 2 & 6 & 9 & 5 & 3 & 1 & 11 \\ 9 & 7 & 2 & 3 & 4 & 5 & 8 & 1 & 11 & 6 & 10 \end{pmatrix}$$

$$p = (1 \ 6 \ 5) (2 \ 4) (3 \ 11 \ 10) (8 \ 9)$$

Aufgabe 3

Stelle die Zyklendarstellung der Permutation der Elemente $1, 2, \dots, 10$ in der Zweizeilenform dar.

$$p = (1 \ 5) (2 \ 9) (3 \ 6 \ 7 \ 4 \ 10)$$

Aufgabe 3

$$p = (1 \ 5) (2 \ 9) (3 \ 6 \ 7 \ 4 \ 10)$$

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 9 & 6 & 10 & 1 & 7 & 4 & 8 & 2 & 3 \end{pmatrix}$$

Aufgabe 4

Wie viele Permutationen aller Elemente einer n -elementigen Menge gibt es

- (a) $n = 4$
- (b) $n = 6$
- (c) $n = 1$
- (d) $n = 0$
- (e) allgemein für n Elemente

Aufgabe 4

- (a) $n = 4 \Rightarrow 4! = 24$ Permutationen
- (b) $n = 6 \Rightarrow 6! = 6 \cdot 5 \cdot 4! = 720$ Permutationen
- (c) $n = 1 \Rightarrow 1! = 1$ Permutation
- (d) $n = 0 \Rightarrow 0! = 1$ Permutation
- (e) allgemein $n \Rightarrow n!$ Permutationen

Aufgabe 5

Notiere alle Permutationen der drei Buchstaben im „Wort“ ASU in lexikographisch aufsteigender Reihenfolge. Wie viele davon haben im Deutschen eine (allgemein) bekannte Bedeutung?

Aufgabe 5

ASU

AUS

SAU

SUA

UAS

USA

Drei (bekannte) Wörter mit Bedeutung im Deutschen.

Aufgabe 6

Ordne die folgenden Permutationen mit dem Symbol „<“ lexikographisch aufsteigend.

$$p_1 = (3, 2, 1, 4, 5)$$

$$p_2 = (4, 2, 1, 5, 3)$$

$$p_4 = (3, 2, 1, 5, 4)$$

$$p_3 = (1, 7, 3, 2, 4)$$

Aufgabe 6

$$p_1 = (3, 2, 1, 4, 5)$$

$$p_2 = (4, 2, 1, 5, 3)$$

$$p_4 = (3, 2, 1, 5, 4)$$

$$p_3 = (1, 7, 3, 2, 4)$$

Aufgabe 6

$$p_1 = (3, 2, 1, 4, 5)$$

$$p_2 = (4, 2, 1, 5, 3)$$

$$p_4 = (3, 2, 1, 5, 4)$$

$$p_3 = (1, 7, 3, 2, 4)$$

$$p_3 < p_1 < p_4 < p_2$$

Aufgabe 7

Bestimme die unmittelbare Nachfolgerin der Permutation $p = (3, 1, 5, 2, 6, 4)$ und notiere die Zwischenresultate der dafür nötigen Teilschritte.

Aufgabe 7

(0)	Ausgangspermutation	(4, 2, 6, 5, 3, 1)
(1)	$i=1$	(4, <u>2</u> , 6, 5, 3, 1)
(2)	$j=4$	(4, <u>2</u> , 6, 5, <u>3</u> , 1)
(3)	Tausche $L[1]$ und $L[4]$	(4, 3, 6, 5, 2, 1)
(4)	reverse $L[2], \dots, L[5]$	(4, 3, 1, 2, 5, 6) = nächste Permutation

Aufgabe 8

Bestimme die unmittelbare Nachfolgerin der Permutation $p = (6, 1, 5, 4, 3, 2)$ und notiere die Zwischenresultate der dafür nötigen Teilschritte.

Aufgabe 8

- | | | |
|-----|-----------------------------|---|
| (0) | Ausgangspermutation | $(6, 1, 5, 4, 3, 2)$ |
| (1) | $i=1$ | $(6, \underline{1}, 5, 4, 3, 2)$ |
| (2) | $j=5$ | $(6, \underline{2}, 5, 4, 3, \underline{1})$ |
| (3) | Tausche $L[1]$ und $L[5]$ | $(6, 2, 5, 4, 3, 1)$ |
| (4) | reverse $L[2], \dots, L[5]$ | $(6, 2, 1, 3, 4, 5) = \text{nächste Permutation}$ |

Aufgabe 9

Vervollständige die Funktion `reverse_slice(L, i, j)`, die in der Liste `L` die Reihenfolge der Elemente `L[i]`, `L[i+1]`, ..., `L[j-1]` inplace umkehrt. Einrückung: 2 Zeichen

[illegible]

Aufgabe 9

```
def reverse_slice(L,i,j):
    while i<j-1:
        L[i],L[j-1]=L[j-1],L[i]
        i+=1
        j-=1
```

Aufgabe 10

Vervollständige die Funktion `next_permutation(L)`, welche die lexikographisch auf L folgende Permutation inline bestimmt.

Verwende dafür die Funktion `reverse_range(L,i,j)`, welche die Elemente `L[i], L[i+1], ..., L[j-1]` inplace umkehrt.

Einrückung: 2 Zeichen

```
def next_permutation(L):
```

Aufgabe 10

```
def next_permutation(L):  
    n=len(L)  
    i=n-2 #Step 1  
    while not(i<0 or L[i]<L[i+1]):  
        i-=1  
    if i<0:  
        return False  
    j=n-1 #Step 2  
    while not(L[i]<L[j]):  
        j-=1  
    L[i],L[j]=L[j],L[i] #Step 3  
    reverse_slice(L,i+1,n) #Step 4  
    return True
```