

1. Du kannst informell oder formal beschreiben, was eine n -stellige Permutation ist.
2. Du kannst die Zweizeilenform einer n -stelligen Permutation interpretieren.
3. Du kannst die Tupelschreibweise („Einzeilenform“) einer n -stelligen Permutation interpretieren.
4. Du kannst die Zyklenschreibweise einer n -stelligen Permutation interpretieren.
5. Du kannst eine Permutation als gerichteten Graphen darstellen.
6. Du kannst die Zweizeilenform in die Tupelschreibweise umformen und umgekehrt.
7. Du kannst die Zyklenschreibweise in die Zweizeilenform umformen und umgekehrt.
8. Du kannst zwei oder mehr Permutationen in Zweizeilenform in der richtigen Reihenfolge verketteten (verknüpfen, komponieren) und kennst den dafür verwendeten Operator „ \circ “.
9. Du weißt, dass die Verknüpfung von Permutationen nicht kommutativ ist, d. h. dass die Identität $p \circ q = q \circ p$ im Allgemeinen *nicht* gilt.
10. Du kannst Potenzen $p^k = p \circ p \circ \dots \circ p$ (k Faktoren) berechnen. Indem du die Periodizität der Permutation p bestimmst, kannst du diese Aufgabe auch für grosse Exponenten k lösen.
11. Du kannst die *identische Permutation* von n Elementen angeben.
12. Du kannst für angemessene Werte von n berechnen, wie viele Nullen am Ende von $n!$ stehen.
13. Du kannst die *inverse Permutation* p^{-1} einer Permutation p bestimmen.
14. Du kannst die Anzahl der *Inversionen* (Fehlstände) einer Permutation berechnen und diese Zahl als Grad für die „Unordnung“ interpretieren, welche durch die Permutation verursacht wird.
15. Du kannst die Anzahl der Permutationen von n Elementen für kleine Werte von n berechnen.
16. Du kannst geeignete Terme mit Fakultäten vereinfachen.
17. Du kannst Permutationsnetzwerke interpretieren, die auch Subnetzwerke mit bekanntem Permutationsschema enthalten können.
18. Du kannst das Vertauschungsschema des Algorithmus von Heap in tabellarischer Form angeben.
19. Du kannst den Algorithmus von Heap für eine beschränkte Anzahl Schritte als Permutationsnetzwerk darstellen.