

Aufgabe 1:

- a) id konsumiert eine Natürliche Zahl und gibt diese wieder zurück.
Es handelt sich um die Identitätsfunktion
- c) f (duplicate_smallest) konsumiert zwei Natürliche Zahlen und gibt das Doppelte der kleineren Zahl zurück.
- e) Damit das lemma $f(x,y) = g(x) \vee f(x,y) = g(y)$ müssen wir $g(x) = f(x,x)$ wählen.
Das heißt wir geben das Doppelte von x zurück.

Aufgabe 3:

Variante 1:

Zählt die Elemente eines Typs (= gleiche Zahl) in der sortierten und unsortierten Liste. Ist die Anzahl gleich und gilt das für alle Typen (alle Zahlen in der Liste) sind die Listen Permutationen.

5 # 1234567890 -> 1
5 # 0987654321 -> 1

Variante 2:

Betrachtet jeweils ein Element der einen Liste und entfernt dieses auch aus der anderen. Dieser Vorgang wird solange wiederholt, bis eine von beiden Listen leer ist. Wenn die Liste leer ist, muss auch die andere leer sein, sonst ist es keine Permutation.

1|234567890 -> 2|34567890 -> 3|4567890 -> ... -> leer
0987654321| -> 098765432| -> 09876543| -> ... -> leer

Variante 3:

Generiert alle möglichen Permutationen einer Liste und überprüft ob die andere (sortierte) Liste in den Permutationen vorhanden ist. Ist das der Fall ist es offensichtlich ein Permutation, anderenfalls eben nicht.

123 € (123, 132, 213, 231, 321, 312) -> Ja

Anderes Sortierverfahren:

Prinzipiell ist es egal, welche Spezifikation man benutzen würde, da alle drei die Permutationseigenschaft beschreiben. Allerdings ist die Variante 3 sehr Speicheraufwendig (es gibt $n!$ Permutationen für eine Liste mit n Elementen). Ich würde mit Variante 2 arbeiten, da diese am leichtesten zu verstehen ist und mit kleiner werdenden Listen arbeitet. Das bedeutet eine geringere Speichernutzung und hohes Verständniss.