

2. táblás gyakorlat – specifikáció, összegzésre visszavezetés

Specifikáld: rendezzünk egy számokat tartalmazó tömböt!

Itt nem tartjuk meg az előfeltételt, hiszen az a cél, hogy a program futásának végére a tömb megváltozzon (rendezve legyen)

$$A = (t: \mathbb{Z}^n)$$

$$ef = (t = t')$$

$$uf = (t \in \text{perm}(t') \wedge \forall i \in [1..n-1]: t[i] \leq t[i+1])$$

- Azért \leq , mert lehetnek a tömbben azonos elemek is.
- Azért $n-1$ -ig, hogy a $t[i+1]$ ne indexeljen ki a tömbből.

$\text{perm}(t')$ a t' tömb összes permutációinak halmazát jelenti, azaz a t' tömb elemeiből képzett összes lehetséges tömböt. Azaz, t egy olyan tömb legyen, aminek az elemei azonosak a kiinduló állapotában vett elemekkel, csak épp a sorrendjük lehet más. Ezt a kifejezést, ha szükséges lesz, a zh-ban lehet majd használni és nem kell külön definiálni.

De azért nézzük csak meg, hogyan lehetne formálisan megadni ezt:

A perm függvény típusa:

$$\text{perm}: \mathbb{Z}^n \rightarrow 2^{\mathbb{Z}^n}$$

Hiszen egy adott tömbhöz (\mathbb{Z}^n) tömbök halmazát rendeli ($2^{\mathbb{Z}^n}$).

De akár így is megadhatnánk a típusát:

$$\text{perm} \subseteq \mathbb{Z}^n \times \mathbb{Z}^n, \text{ úgy hogy: } \text{dom}(\text{perm}) = \mathbb{Z}^n$$

A dom jelentése értelmezési tartomány, azaz perm egy olyan *reláció*, ami minden \mathbb{Z}^n -beli tömbhöz rendel elemet, de egyhez akár többet is.

És akkor most kíséreljük meg megadni a függvény pontos jelentését (az első, halmazos szignatúrából indulok ki):

$$\forall t \in \mathbb{Z}^n: \text{perm}(t) := \{s \in \mathbb{Z}^n \mid \forall i \in [1..n]: \exists j \in [1..n]: t[i] = s[j]\}$$

Ez nem jó... Azért nem jó, mert nézzük az alábbi példát:

$$t := [1,1,2]$$

Ekkor

$$\text{perm}(t) = \{[1,1,2], [1,2,1], [2,1,1]\}$$

De a fenti definíció nem ezt a halmazt adja meg, hanem egy ennél bővebbet, hiszen pl. a $[1,2,2]$ tömb minden eleméhez is lesz olyan index, ahol t pont azt az elemet veszi fel, ez mégse permutációja t -nek!

Viszont azt se mondhatom, hogy legyen ez a definíció:

$$\forall t \in \mathbb{Z}^n: \text{perm}(t) := \{s \in \mathbb{Z}^n \mid \forall i \in [1..n]: \exists! j \in [1..n]: t[i] = s[j]\}$$

Mert ha van több azonos érték t -ben (mint ahogy a példában van is), akkor ez a definíció ellentmondásos, hiszen azt várja el, hogy mindkét t -beli 1-es értékhez ugyanaz az 1-es tartozzon, miközben a permutált tömbnek is 3 eleműnek kéne lennie.

Ez abban az esetben egyébként teljesen jó megoldás, ha tudjuk hogy a tömbben nincsenek azonos elemek (de akkor már az előző megoldás is jó).

Ezzel szemben a helyes, általános érvényű megoldás így néz ki:

$$\forall t \in \mathbb{Z}^n: \text{perm}(t) :=$$

$$\{s \in \mathbb{Z}^n \mid \exists \text{bij}: [1..n] \rightarrow [1..n]: (\forall i \in [1..n]: \forall j \in [1..n]: (\text{bij}(i) = \text{bij}(j) \rightarrow i = j) \wedge \forall i \in [1..n]: s[i] = t[\text{bij}(i)])\}$$

Tehát olyan s tömbök alkotják a permutációk halmazát, amelyekhez létezik olyan az indextartományon értelmezett *bijekció*, amit ráeresztve t egyik indexére épp megkapjuk s egyik elemét a t tömbben. És mivel ez minden indexre igaz, ezért t és s egymás permutáltjai.

Gyakorlatilag azért kellett ez a bonyolítás, mert az értékek között ugyan lehetnek egyformák, de az indexek között nem.

Azt az értéket (vagy érték n -est), ami egy adathalmaz bármely elemét *egyértelműen* azonosít a halmaz többi eleméhez képest, *kulcsnak* szokás hívni. Az emberek halmazában a név nem, de pl. a személyi szám (vagy újabban a személyiigazolvány-szám) egy ilyen kulcs. Ebben a példában a tömbindex a kulcs. Ezt a fogalmat majd *adatbázisok* tárgyból fogjátok használni.