

Třídění zboží ve skladu

Ve skladu se nachází několik kontejnerů, každý obsahuje naskládané krabice s různými typy produktů. Každá krabice obsahuje právě jeden typ produktu. Napište program `warehouse.py`, který uspořádá krabice mezi kontejnery tak, aby byl dosažen požadovaný cílový stav skladu s **minimálním počtem přesunů**, přičemž musí být dodržena určitá pravidla týkající se bezpečnosti a kapacity kontejnerů.

Pravidla přesunu

1. V jednom kroku lze přesunout pouze jednu krabici.
2. Krabici lze umístit pouze na horní pozici jiného kontejneru.
3. Kontejner má omezenou kapacitu 4 krabice.
4. Pokud je kontejner plný, nelze do něj přidat další krabici.

Bezpečnostní pravidla

Ve skladu se skladují tyto typy produktů: **Chemikálie**, **Elektronika**, **Kyseliny**, **Kovy**, **Potraviny**, **Sklo**, **Hořlaviny**. Pro které platí následující bezpečnostní pravidla:

- **Nekompabilní produkty se nikdy nesmí skladovat ve stejném kontejneru:**
 - Chemikálie \neq Potraviny
 - Hořlaviny \neq Elektronika
 - Kyseliny \neq Kovy
- **Těžké produkty** (Kovy) nesmí být položeny na **křehké produkty** (Sklo).

Vstup / Výstup

- **Vstup:** JSON soubor s klíči `start` a `target` (seznamy kontejnerů a krabic) na příkazové řádce. Pro načtení vstupu můžete použít následující kód:

```
import json
with open(filename, "r", encoding="utf-8") as f:
    data = json.load(f)
```

kde v proměnné `data` bude slovník s klíči `start` a `target`, obsahující počáteční a cílový stav skladu.

- **Výstup:** jeden řádek obsahující posloupnost přesunů krabic ve formátu `source destination`, kde `source` a `destination` jsou indexy kontejnerů mezi kterými se krabice přesouvá. Např. `0 2 1 2` popisuje dva přesuny: nejprve z kontejneru 0 do 2, poté z kontejneru 1 do 2.

Příklady

Příklad 1

Vstupní soubor:
`sklad1.json`

```
{ "start": [
  [ "Chemikalie", "Chemikalie", "Chemikalie", "Elektronika" ],
  [ "Kyseliny", "Kyseliny" ],
  [ "Elektronika", "Elektronika", "Elektronika" ],
  [ "Kyseliny", "Chemikalie", "Kyseliny" ]
],
"target": [
  [ "Chemikalie", "Chemikalie", "Chemikalie", "Chemikalie" ],
  [ "Kyseliny", "Kyseliny", "Kyseliny", "Kyseliny" ],
  [ "Elektronika", "Elektronika", "Elektronika", "Elektronika" ],
  []
] }
```

Výstup:

```
0 2 3 1 3 0 3 1
```

Vizualizace řešení je níže. Správné řešení je i 3 1 0 2 3 0 3 1.

Příklad 2

Vstupní soubor:

sklad2.json

```
{ "start": [
  [ "Sklo", "Kyseliny", "Potraviny", "Potraviny" ],
  [ "Potraviny", "Sklo"],
  [ "Kyseliny", "Kyseliny" ],
  [ "Kyseliny", "Sklo", "Sklo", "Potraviny" ]
],
"target": [
  [ "Sklo", "Sklo", "Sklo", "Sklo" ],
  [ "Potraviny", "Potraviny", "Potraviny", "Potraviny" ],
  [ "Kyseliny", "Kyseliny", "Kyseliny", "Kyseliny" ],
  []
] }
```

Výstup:

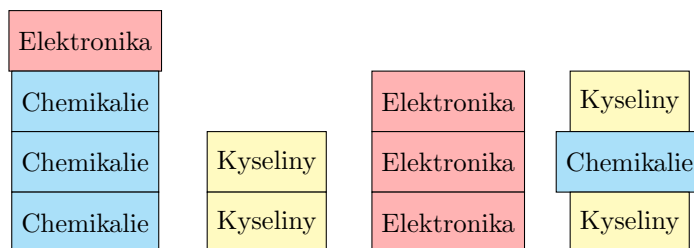
```
1 2 0 1 0 1 0 1 2 0 1 2 3 1 3 0 3 0 3 2
```

Správná řešení jsou i:

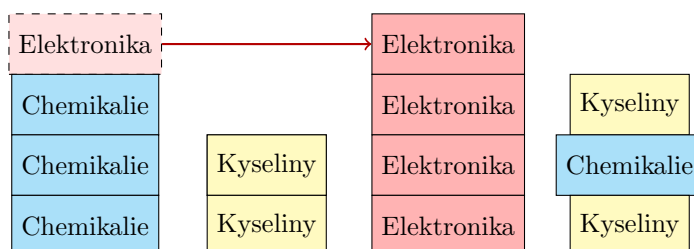
```
1 2 0 1 0 1 2 1 0 2 1 0 3 1 3 0 3 0 3 2,
1 2 0 1 0 1 3 1 0 3 2 0 3 2 3 0 3 0 3 2,
1 2 0 1 0 1 3 1 2 3 0 2 3 0 3 0 3 0 3 2,
1 2 0 1 3 1 0 1 0 3 2 0 3 2 3 0 3 0 3 2,
1 2 0 1 3 1 0 1 2 3 0 2 3 0 3 0 3 0 3 2,
1 2 0 1 3 1 2 3 0 1 0 2 3 0 3 0 3 0 3 2,
1 2 3 1 0 1 0 1 0 3 2 0 3 2 3 0 3 0 3 2,
1 2 3 1 0 1 0 1 2 3 0 2 3 0 3 0 3 0 3 2,
1 2 3 1 0 1 2 3 0 1 0 2 3 0 3 0 3 0 3 2,
1 2 3 1 2 3 0 1 0 1 0 2 3 0 3 0 3 0 3 2,
3 2 1 3 0 1 0 1 2 1 0 2 3 0 3 0 3 0 3 2,
3 2 1 3 0 1 2 1 0 1 0 2 3 0 3 0 3 0 3 2,
3 2 1 3 2 1 0 1 0 1 0 2 3 0 3 0 3 0 3 2.
```

Vizualizace řešení příkladu 1

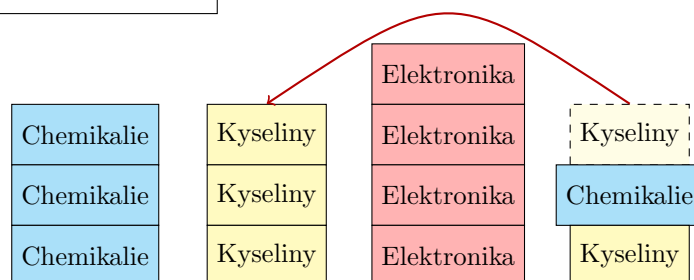
Krok 0: start



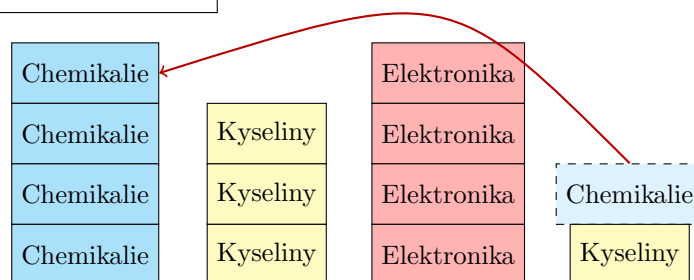
Krok 1: Přesun 0→2



Krok 2: Přesun 3→1



Krok 3: Přesun 3→0



Krok 4: Přesun 3→1 (target)

