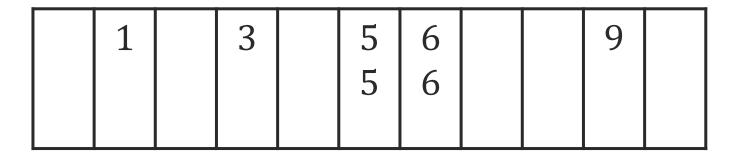
# ADATSZERKEZETEK ÉS ALGORITMUSOK

- Tegyük fel, hogy tudjuk, hogy a bemenő elemek (A[1..n] elemei) egy m elemű U halmazból kerülnek ki
  - Például  $\forall A[i]$ -re igaz, hogy  $A[i] \in [1..m]$ .
- Lefoglalunk egy U elemeivel indexelt B tömböt (m db ládát), először mind üres
- B segédtömb elemei lehetnek láncolt listák például

- Első fázis: végigolvassuk az A-t, és az s = A[i] elemet a B[s] lista végére fűzzük.
- Például tegyük fel, hogy a rendezendő A[1..7] tömb elemei 0 és 10 közötti egészek
- A:

• B:

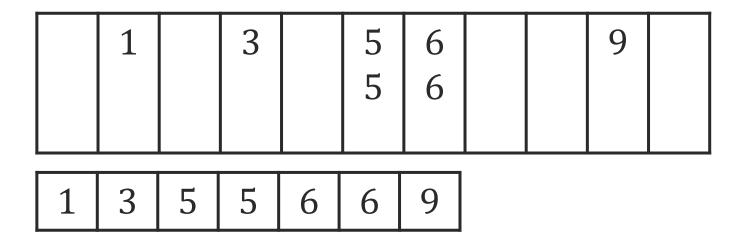
5	3	1	5	6	9	6
---	---	---	---	---	---	---



• Második fázis: elejétől a végéig növő sorrendben végigmegyünk B-n, és a B[i] listák tartalmát visszaírjuk A-ba.

• B:

• A



- Lépésszám:
  - B létrehozása: O(m),
  - első fázis  $\mathcal{O}(n)$ ,
  - második fázis  $\mathcal{O}(n+m)$ , összesen  $\mathcal{O}(n+m)$ .
- Ez gyorsabb, mint az általános alsó korlát, ha:
  - $m \leq c * n$

- Általánosabban:
  - Legyen K az S sorozat elemeinek típusértékhalmaza,  $\varphi \colon K \to [0 \dots M-1]$  olyan függvény, amire igaz, hogy ha  $\varphi(k_1) < \varphi(k_2)$ , akkor  $k_1 < k_2$ .
  - Legyenek  $E_0, E_1, \dots E_{M-1}$  edények, melyek éppen olyan sorozatok, mint S.
  - Az egyes edényekben megmarad az S-beli elemek ottani relatív sorrendje.

Edenyrendezo(S)

```
E_0, E_1, \dots, E_{M-1} \leftarrow \varepsilon, \varepsilon, \dots, \varepsilon S \neq \varepsilon Out(S, x) In(E_{\varphi(x)}, x) S \leftarrow \ddot{\text{Osszefűz\'es}}(E_0, E_1, \dots E_{M-1})
```

- Ahhoz, hogy Edenyrendezo(S) egyszeri szétrakással rendezzen, elégséges, ha:
  - minden edényben legfeljebb egy elem van,  $(\forall i \in [0, M-1]: |E_i| \leq 1)$  vagy:
  - az egyes edényekben csak azonos elemek vannak, vagy:
  - az egyes edények rendezettek.
- Ha az előző feltételek valamelyike teljesül, akkor tökéletes az edényrendezés.
  - Ennek műveletigénye:  $\Theta(n)$ , ahol n = |S|.
  - Példa: nagyon sok embert kell magasság szerint sorba rakni
    - megéri minden egyes testmagasság számára egy edényt létrehozni

- Tegyük fel, hogy a kulcsok összetettek, több komponensből állnak,  $(t_1, \dots t_k)$  alakú szavak, ahol a  $t_i$  komponens az  $L_i$  rendezett típusból való, a rendezés lexikografikus
- Példa: Legyen (U; <) a huszadik századi dátumok összessége az időrendnek megfelelő rendezéssel:

```
• L_1 = \{1900; 1901; ...; 1999\} n_1 = 100
• L_2 = \{\text{január, február, ..., december}\} n_2 = 12
• L_3 = \{1; 2; ...; 31\} n_3 = 31
```

• A dátumok rendezése éppen az  $L_i$  típusokból származó lexikografikus rendezés lesz

- 1. Rendezzük a sorozatot az utolsó, a *k*-adik komponensek szerint edényrendezéssel!
- 2. A kapott eredményt rendezzük a k-1-edik komponensek szerint edényrendezéssel, stb.
- Fontos, hogy az edényrendezésnél az elemeket a ládában mindig a lista végére tettük.
  - Így, ha két azonos kulcsú elem közül az egyik megelőzi a másikat, akkor a rendezés után sem változik a sorrendjük ⇒ stabil rendezés.

- Miért működik a radix jól?
  - Ha X < Y, az első i-1 tag megegyezik, de  $x_i < y_i$ , akkor az i-edik komponens rendezésekor X előre kerül.
  - Stabil rendezés ⇒ később már nem változik a sorrendjük

#### Példa:

```
1969. jan. 18. 1969. jan. 1. 1955. dec. 18. 1955. jan. 18. 1918. dec. 18. 1969. jan. 1. 1969. jan. 18. 1955. dec. 18. 1955. jan. 18. 1918. dec. 18. 2. menet után:

1969. jan. 1. 1969. jan. 18. 1955. jan. 18. 1955. dec. 18. 1918. dec. 18. 3. menet után:

1918. dec. 18. 1955. jan. 18. 1955. dec. 18. 1969. jan. 1. 1969. jan. 18.
```

Helytelen a sorrend

Példa – ha fordítva lenne

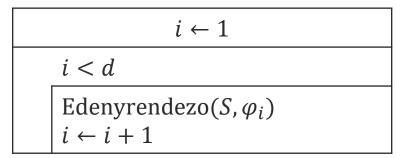
```
1969. jan. 18. 1969. jan. 1. 1955. dec. 18. 1955. jan. 18. 1918. dec. 18. 1. menet után: 1918. dec. 18. 1955. dec. 18. 1955. jan. 18. 1969. jan. 18. 1969. jan. 18. 1969. jan. 1. 2. menet után: 1955. jan. 18. 1969. jan. 18. 1969. jan. 1. 1918. dec. 18. 1955. dec. 18. 3. menet után: 1969. jan. 1. 1955. jan. 18. 1969. jan. 18. 1918. dec. 18. 1955. dec. 18.
```

#### Általánosan

- Az  $e=e_de_{d-1}\dots e_2e_1$  számot jobbról balra, az alacsony helyiértékek felől indulva pozíciónként szétrakja edényekbe, majd összefűzi az edények tartalmát
- Az i. pozíción a  $\varphi_i$  függvényt alkalmazzuk:  $\varphi_i(e) = e_i$
- Az i. pozíción végrehajtott szétrakás és összefűzés után S "i-rendezett" lesz.

- Definíció
  - S "i-rendezett" (jelölés:  $x \leq_i y$ ), ha minden
    - $x = x_d x_{d-1} \dots x_2 x_1$  és  $y = y_d y_{d-1} \dots y_2 y_1$ -ra:  $x <_0 y$
    - $x \le_i y \Leftrightarrow x_i < y_i \text{ vagy } x_i = y_i \text{ és } x <_{i-1} y \ (i > 0)$
  - Ekkor a "d-rendezés" a közönséges rendezés
- Hatékonyság:
  - 2 \* d-szer megyünk végig az S sorozaton, így  $T(n) = \Theta(d*|S|)$

Radix(S)



- Szokásos implementáció:
  - S fejelemes láncolt lista
  - Az edényeket egy "fej" és egy "vége" mutató ábrázolja.
  - A szétrakás és az összefűzés az elemek láncolásával megoldható.
  - Összefűzéskor nem kell az egyes edények részlistáit végigolvasni, hanem egy darabban lehet őket láncolni.

# Edényrendező implementáció

Következő téma