

2. tétel

Adatszerkezetek és algoritmusok vizsga

Frissült: 2013. január 28.

Tömbök

Definíció ADT szinten

Az E alaptípusú k dimenziós ($k \geq 1$) T tömbtípus. legyen

$$I = I_1 \times I_2 \times \dots \times I_k$$

egy indexhalmaz, ahol

$$\forall j \in [1, k] : I_j = [1, n_j]$$

(kezdőérték lehetne m_j is).

Az $A \in T$ tömbnek

$$N = n_1 * n_2 * \dots * n_k$$

eleme van:

$$\{a_1, a_2, \dots, a_k\}$$

Mindig van egy $f : I \rightarrow \{a_1, a_2, \dots, a_N\}$ egy-egy értelmű leképezés.

Jelölés: $A[i_1, i_2, \dots, i_k]$ a tömbnek az $i_1, i_2, \dots, i_k$ indexek által azonosított eleme.

Invariáns

Speciális megszorítás. Pl. legyen szimmetrikus, alsó háromszög, ritka stb.

Műveletek

- Indexelés: az $i_1, i_2, \dots, i_k$ indexhez tartozó $A[i_1, i_2, \dots, i_k]$ elem kiválasztása.
- Elem módosítás: $A[i_1, i_2, \dots, i_k] := a$ - értékadás
- $A := B$ értékadás
- stb.

Elnevezések

Ha $k = 1$, akkor **vektor**, ha $k = 2$, akkor **mátrix**.

Tömbök ADS szinten

ADS szinten nem kötelező szerkezetet, rákövetkezést definiálni az elemek közt, de elfogadott a **köv_j** reláció bevezetése $\forall j \in [1, k]$ -ra:

$$\text{köv}_j(A[i_1, \dots, i_j, \dots, i_k]) = A[i_1, \dots, i_{j+1}, \dots, i_k]$$

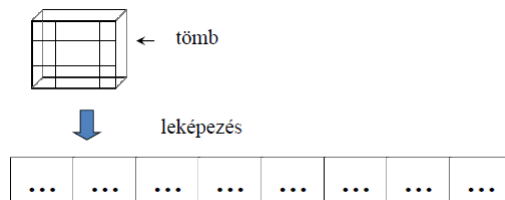
ha $i_j < n_j$, egyébként **köv_j** nem definiált.

Egy belső elemnek minden dimenzióban van rákövetkezője.

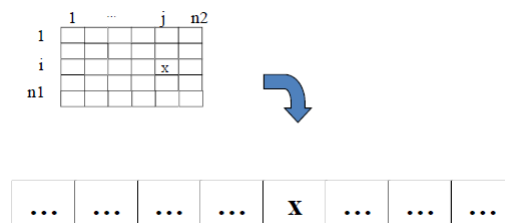
A legalább 2 dimenziós tömböt **ortogonális adatszerkezetnek** nevezzük.

Reprezentáció

Tömbök aritmetikai ábrázolása. Egy k -dimenziós tömböt szeretnénk elhelyezni egy alkalmas méretű 1-dimenziós tömbben (vektorban), és megadjuk a leképezés index- vagy címfüggvényét.



Példa: mátrix elhelyezése vektorban:



Elhelyezés ált. lehet **sorfolytonos** vagy **oszlopfolytonos**.

Sorfolytonos ábrázolás

Indexfüggvény

$\forall i \in [1, n_1], \forall j \in [1, n_2]$:

$$\text{ind}(A[i, j]) = (i - 1) * n_2 + j$$

Más jelölés: szokták úgy is tekinteni, hogy a mátrixnak m sora és n oszlopa van. $\forall i \in [1, m], \forall j \in [1, n]$

$$\text{ind}(A[i, j]) = (i - 1) * n + j$$

Címfüggvény

$$\text{cím}(A[i, j]) = \text{cím}(A) + (i - 1) * n_2 * h + (j - 1) * h$$

$$\text{cím}(A[i, j]) = \text{cím}(A) + (i - 1) * n * h + (j - 1) * h$$

$\text{cím}(A)$: kezdőcím, h : egy elem hossza.

Oszlopfolytonos ábrázolás

Indexfüggvény

$$\text{ind}(A[i, j]) = (j - 1) * n_2 + i$$

$$\text{ind}(A[i, j]) = (j - 1) * m + i$$

Címfüggvény

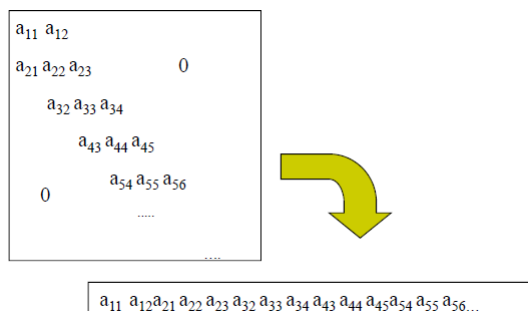
$$\text{cím}(A[i, j]) = \text{cím}(A) + (j - 1) * n_2 * h + (i - 1) * h$$

$$\text{cím}(A[i, j]) = \text{cím}(A) + (j - 1) * m * h + (i - 1) * h$$

Speciális mátrixok

Cím, index függvényei, hézagos mátrixok reprezentációja.

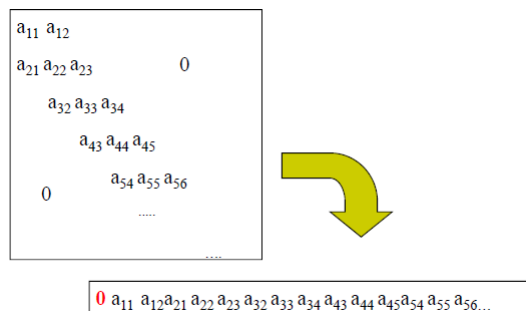
Tridiagonális mátrix



Ha $|i - j| \leq 1$, akkor

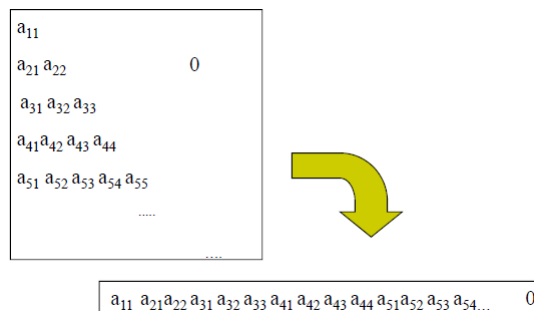
$$\text{ind}(A[i, j]) = (i - 1) * 3 - 1 + \begin{cases} 1 & \text{ha } i > j \\ 2 & \text{ha } i = j \\ 3 & \text{ha } i < j \end{cases}$$

Ha a '0' elemet is tároljuk pl. a célvektor elején:



$$\text{ind}(A[i, j]) = \begin{cases} (i - 1) * 3 + 1 & \text{ha } i = j + 1 \\ (i - 1) * 3 + 2 & \text{ha } i = j \\ i * 3 & \text{ha } i + 1 = j \\ 1 & \text{különben} \end{cases}$$

Alsó háromszög mátrix



elemek száma:

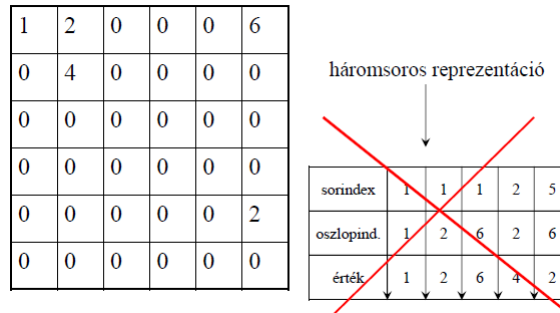
$$m = n * (n + 1) / 2 + 1$$

indexfüggvény:

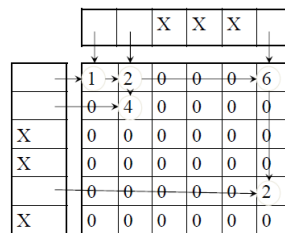
$$\text{ind}(A[i, j]) = \begin{cases} i * (i - 1) / 2 + j & \text{ha } i \geq j \\ n * (n + 1) / 2 + 1 & \text{ha } i < j \end{cases}$$

Hézagosan kitöltött mátrixok

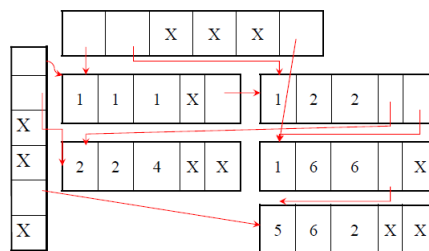
Lehetne így is reprezentálni, de nem így tesszük:



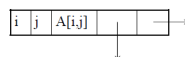
hanem láncolt / vegyes ábrázolást használunk.
Szemléletes ábra:



A "valóságban", a memóriában így jelennek meg:



Egy darab adatalem pedig így néz ki:



minden adatalemnek van egy mellette és egy alatta levőre mutatója (ami nem feltétlenül közvetlenül mellette/alatta van!), illetve a sor és oszlop "fejlécek" elemenként az a sorban/oszlopban legelső elemre mutatnak (ha van).

Ez ugyanis sokkal tömörebb. Legyen egy $m * n$ -es mátrixunk. Ennek a nem nulla elemeinek száma k . Egy db. érték helyfoglalása h byte. Egy mutató p byte helyet foglal el, egy index i byte-ot. A hely amit lefoglalunk:

$$(h + 2 * i + 2 * p) * k + (m + n) * p \ll m * n * h$$

lényegesen kisebb mint a háromsoros reprezentációnál (bár bonyolultabb)