Adatszerkezetek és algoritmusok

Horváth Géza

harmadik előadás

Előadások témái

- Az algoritmusokkal kapcsolatos alapfogalmak bevezetése egyszerű példákon keresztül.
- Az algoritmusok futási idejének aszimptotikus korlátai.
- Az adatszerkezetekkel kapcsolatos alapfogalmak. A halmaz, a multihalmaz és a tömb adatszerkezet bemutatása.
- Az adatszerkezetek folytonos és szétszórt reprezentációja. A verem, a sor és a lista.
- Táblázatok, önátrendező táblázatok, hash függvények és hash táblák, ütközéskezelés.
- o Fák, bináris fák, bináris keresőfák, bejárás, keresés, beszúrás, törlés.
- Kiegyensúlyozott bináris keresőfák: AVL fák.
- Piros-fekete fák.
- B-fák.
- Gráfok, bejárás, legrövidebb út megkeresése.
- Párhuzamos algoritmusok.
- Eldönthetőség és bonyolultság, a P és az NP problémaosztályok.
- Lineáris idejű rendezés. Összefoglalás.

Adatszerkezet: adatelemek + kapcsolatok + műveletek + tárolás (reprezentáció).

Tehát az adatszerkezet adatokból, a köztük lévő kapcsolatokból, a rajtuk végrehajtható műveletekből és a memóriában illetve a háttértáron történő fizikai tárolás megvalósításából áll.

- Az absztrakt adattípus az adatelemek és a köztük lévő kapcsolatok logikai modellje. Független az adattárolás fizikai megvalósításától.
- Az adatszerkezet mindezeken túl tartalmazza az adatok fizikai tárolásának a memóriában/háttértáron történő megvalósítását és az adatokat manipuláló műveletek megvalósítását is.

Műveletek adatszerkezetekkel

Műveletek:

- adatszerkezetek létrehozása
- adatszerkezetek módosítása
 - elem hozzáadása
 - elem cseréje
 - elem törlése
- elem elérése

Néha: rendezés, keresés, bejárás, feldolgozás

Adatszerkezetek reprezentációja

A mai modern számítógépek közvetlen elérésű memóriával rendelkeznek. (RAM random-access memory). Ez azt jelenti, hogy a memóriában tárolt minden elem azonos (konstans) idő alatt hozzáférhető, függetlenül attól, hogy az adott elem elérésekor hány elemet tartalmaz a memória.

Az adatszerkezet reprezentációja lehet:

- folytonos (vektorszerű) ma
- Szétszórt (láncolt) jövő héten

Adatszerkezetek osztályozása

- az adatszerkezet elemeinek típusa alapján
 - homogén (azonos típusú elemekből áll)
 - heterogén (különböző típusú elemekből áll)
- az adatszerkezet elemeinek száma alapján
 - dinamikus (változhat az elemek száma)
 - statikus (az elemek száma fix)
- a homogén adatszerkezet elemeinek kapcsolata alapján
 - struktúra nélküli
 - asszociatív
 - szekvenciális
 - hierarchikus
 - hálós

Példa – bináris keresés

A **bináris keresés** művelet megvalósításához elengedhetetlenek a következők:

- homogén adatszerkezet (azonos típusú és méretű elemek)
- rendezett adatok
- folytonos reprezentáció (az adatok közvetlen elérésével)

A halmaz, mint absztrakt adattípus

A halmaz egymástól megkülönböztethető elemekből áll. Ha x az S halmaz eleme, azt így jelöljük: $x \in S$. Ha x nem eleme az S halmaznak, azt így jelöljük: $x \notin S$. A halmaz elemeit felsorolással is megadhatjuk, például ha az S halmaz az 1, 2 és 3 számokból áll, azt így jelöljük: $S = \{1,2,3\}$.

Legyenek adva az A és B halmazok. Ekkor végrehajthatóak az alábbi halmazműveletek.

Halmazműveletek:

• Az A és a B metszete (vagy közös része) az

$$A \cap B = \{x : x \in A \text{ \'es } x \in B\}$$

halmaz.

• Az A és a B uniója (vagy egyesítése) az

$$A \cup B = \{x : x \in A \text{ vagy } x \in B\}$$

halmaz.

• az A és a B halmaz különbsége az

$$A - B = \{x : x \in A \text{ \'es } x \notin B\}$$

halmaz.

A halmaz, mint absztrakt adattípus

Halmazműveletek tulajdonságai:

Null-elem tulajdonság:

$$A \cap \emptyset = \emptyset,$$

$$A \cup \emptyset = A.$$

Idempotencia:

$$A \cap A = A,$$

$$A \cup A = A.$$

Kommutativitás:

$$A \cap B = B \cap A,$$

$$A \cup B = B \cup A.$$

A halmaz, mint absztrakt adattípus

Asszociativitás:

$$A \cap (B \cap C) = (A \cap B) \cap C,$$

 $A \cup (B \cup C) = (A \cup B) \cup C.$

Disztributivitás:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C),$$

 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$

Elnyelés:

$$A \cap (A \cup B) = A,$$

 $A \cup (A \cap B) = A.$

DeMorgan-azonosságok:

$$A - (B \cap C) = (A - B) \cup (A - C),$$

$$A - (B \cup C) = (A - B) \cap (A - C).$$

A halmaz, mint adatszerkezet

A halmaz tulajdonságai:

- homogén
- dinamikus
- struktúra nélküli
- folytonos reprezentációval ábrázolt

A halmaz reprezentációja

Legyen az A halmaz a 0 és 9 közötti egész számok egy részhalmaza.

Például $A = \{0,3,4,5,6,8\}$.

Az A halmaz jó reprezentációjához mindössze 10 bit szükséges:

Halmazműveletek megvalósítása

Legyenek az A és B halmazok a 0 és 9 közötti egész számok részhalmazai.

Például A= $\{0,3,4,5,6,8\}$, B= $\{1,3,5,9\}$.

Az A halmaz reprezentációja:

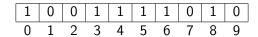
1	0	0	1	1	1	1	0	1	0
0	1	2	3	4	5	6	7	8	9

A B halmaz reprezentációja:

0	1	0	1	0	1	0	0	0	1
0	1	2	3	4	5	6	7	8	9

$$A = \{0,3,4,5,6,8\}, B = \{1,3,5,9\}.$$

Az A halmaz reprezentációja:



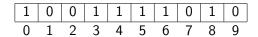
A B halmaz reprezentációja:

Az A és B halmazok **uniójának** kiszámításához a logikai (bitenkénti) **vagy** műveletet használjuk:

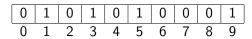
A metszet művelet megvalósítása

$$A = \{0,3,4,5,6,8\}, B = \{1,3,5,9\}.$$

Az A halmaz reprezentációja:



A B halmaz reprezentációja:

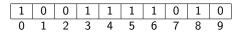


Az A és B halmazok metszetének kiszámításához a logikai (bitenkénti) **és** műveletet használjuk:

A különbség művelet megvalósítása

 $A = \{0,3,4,5,6,8\}, B = \{1,3,5,9\}, \bar{B} = \{0,2,4,6,7,8\}.$

Az A halmaz reprezentációja:



A B halmaz reprezentációja:

Az A és B halmazok különbségének kiszámításához a logikai (bitenkénti) A és nem B műveleteket használjuk. Először számoljuk ki a nem B értékeket:

Majd pedig az **A és nem B** értékeket:

Műveletek halmazokkal, mint adatszerkezetekkel

Műveletek:

- adatszerkezetek létrehozása: folytonos reprezentációval
- adatszerkezetek módosítása
 - elem hozáadása: unió (egyelemű halmaz hozzáadása)
 - elem törlése: különbség (egyelemű halmaz kivonása)
 - elem cseréje: különbség, majd unió
- elem elérése: ∈

A multihalmaz, mint absztrakt adattípus

A matematikában a multihalmaz a halmaz olyan módosítása, melyben azonos elemek többször is előfordulhatnak.

Például az S multihalmaz legyen az a multihalmaz, amely az alábbi számjegyekből áll:

2 db. 1-es, 1 db. 2-es, és 2 db. 3-as. Azaz S={1,1,2,3,3}.

A multihalmazok műveletei és a multihalmazok reprezentácói a gyakorlatokon kerülnek bemutatásra.

A tömb, mint absztrakt adattípus

A tömb olyan adattípus, mely előre meghatározott, konstans számú elemből áll, és minden elemet egyértelműen azonosít egy egész számokból álló véges sorozat. Ezen sorozat elemeinek száma a tömb dimenziója.

Az egydimenziós tömböt vektornak hívjuk, és a számot, amely az adott elem tömbön belüli helyzetét mutatja, az adott elem indexének nevezzük.

Példa: Legyen az A vektor: $A = \{2,11,21,8,6,5,9,3,15,4\}$.

Ebben a példában az A tömb 10 elemből áll, és a 15-ös szám a 9. pozícióban helyezkedik el, ezért a 15 indexe a 9, azaz A[9] = 15.

A kétdimenziós tömböt mátrixnak hívjuk, és minden elemet 2 számjegyből álló sorozattal azonosítunk.

Példa: Legyen M a következő mátrix:

	1	2	3	4	5	6
1	6	11	8	5	22	3
2	12	9	2	21	6	9
3	1	5	24	2	9	17

Ekkor az M mátrix 3 sorból és 6 oszlopból áll, és a 21-es szám 2 számjegyből álló sorozattal, a (2,4)-el azonosítható, mivel a 21 a 2. sorban és 4. oszlopban helyezkedik el. (M[2,4]=21.)

A tömb, mint absztrakt adattípus

Természetesen léteznek magasabb dimenziószámú tömbök is, például a 4 dimenziós tömb esetén 4 számjegyből álló sorozat azonosít minden elemet.

Hogyan lehet elképzelni és kezelni magasabb dimenziószámú tömböket?

A tömb, mint adatszerkezet

A tömb tulajdonságai:

- homogén
- statikus
- asszociatív
- folytonos reprezentációval ábrázolt

A vektor reprezentációja

A vektor ábrázolását tekintjük a legtermészetesebb folytonos reprezentációnak.

Példa:

int A[5];

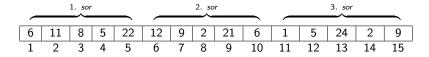
Lefoglalunk 5 egymást követő pozíciót a memóriában egész számok tárolására.

A mátrix reprezentációja – sorfolytonosan

Legyen M a következő mátrix:

	1	2	3	4	5
1	6	11	8	5	22
2	12	9	2	21	6
3	1	5	24	2	9

Ekkor az M mátrix sorfolytonos reprezentációja:



Tárolni kell továbbá a mátrix méretét: (3,5).

Egy mátrix sorfolytonos reprezentációja esetén megmarad az elemekhez történő közvetlen hozzáférés.

A mátrix reprezentációja – oszlopfolytonosan

Legyen M a következő mátrix:

	1	2	3	4	5
1	6	11	8	5	22
2	12	9	2	21	6
3	1	5	24	2	9

Ekkor az M mátrix oszlopfolytonos reprezentációja:

 oszlop 			oszlop			oszlop			oszlop			oszlop		
		$\overline{}$			$\overline{}$		$\overline{}$			$\overline{}$				
6	12	1	11	9	5	8	2	24	5	21	2	22	6	9
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Tárolni kell továbbá a mátrix méretét: (3,5).

Egy mátrix oszlopfolytonos reprezentációja esetén megmarad az elemekhez történő közvetlen hozzáférés.

Ritka mátrix

Ritka mátrixnak nevezünk egy mátrixot, ha a mátrixban elhelyezkedő elemek nagyrészének az értéke 0.

Például legyen M az alábbi ritka mátrix:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
											0			
2	0	0	0	0	0	0	11	0	0	0	0	0	9	0
3	0	5	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	25	0	0	0

Hogyan tudjuk hatékonyan tárolni?

Ritka mátrix – 3 soros reprezentáció

Legyen M az alábbi ritka mátrix:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	3	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	11	0	0	0	0	0	9	0
3	0	5	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	25	0	0	0

Ekkor az M ritka mátrix 3 soros reprezentációja:

sor	1	2	2	3	4
oszlop	4	7	13	2	11
érték	3	11	9	5	25

Tárolni kell továbbá a mátrix méretét: (4,14).

A 3 soros reprezentáció alkalmazásával elveszik az elemekhez történő közvetlen hozzáférés.

Irodalomjegyzék

