A számítástudomány alapjai

Katona Gyula Y.

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Adatszerkezetek

Adatszerkezetek

Definíció

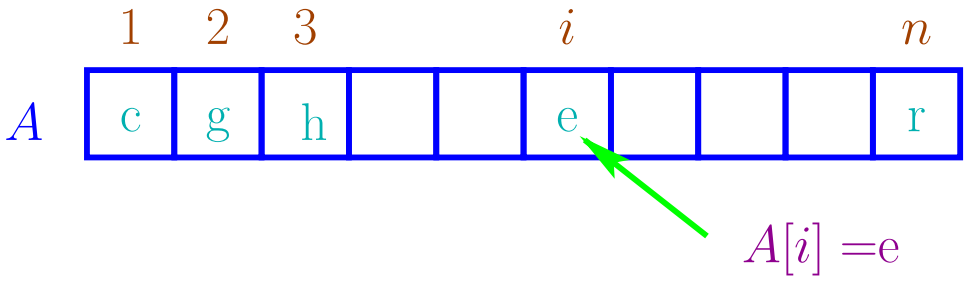
Egy adatszerkezet (data structure) elemek egy halmazának tárolása, az elemek közötti kapcsolat módja és az elemek kezeléséhez tartozó műveletek összessége.

- Vizsgáljuk az elméleti különbségeket a különböző adatszerkezetek között.
- Technikai részletekkel nem nagyon foglalkozunk, az majd lesz programozásból.
- Különböző feladatokhoz különböző adatstruktúra lehet jó, néha érdemes speciális, új adatstruktúrát kitalálni.

Tömb

Definíció

A tömb (array) elemek egy halmazának adott sorrendben való tárolása. Az elemekre az indexükkel lehet hivatkozni. A műveletek: olvas[i], ír[i]



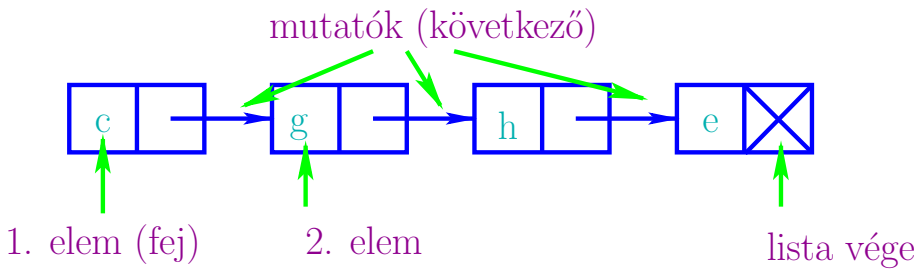
Előnyök: Gyors hozzáférés az elemekhez (konstans idő)

Hátrány: Dinamikus adatoknál nehéz lehet a tömb méretének változtatása, beszúrás, törlés

Láncolt lista

Definíció

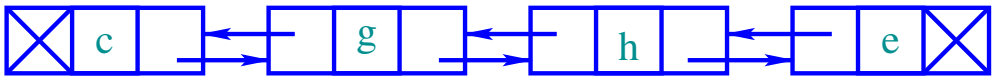
A tömb (linked list) elemek egy halmazának adott sorrendben való tárolása. Olyan csomópontok, cellák sorozatából épül fel, amelyek tetszőleges számú és fajtájú adatmezőt, és egy (vagy két) hivatkozást tárolnak. A hivatkozás(ok) a lista következő (és előző) elemére mutat(nak). A műveletek: első_elem, aktuális_elem, következő_elem, beszúrás, törlés, (előző_elem)



Előnyök: Gyors beszúrás, törlés (konstans idő)

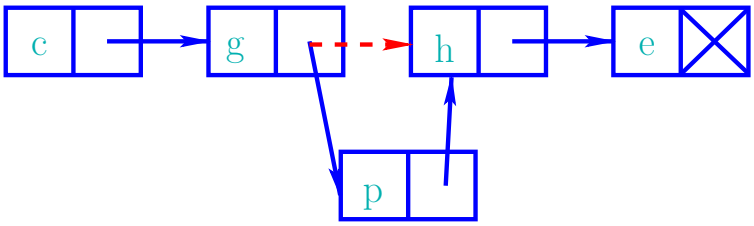
Hátrány: Egyes elemekhez való hozzáférés, keresés lassabb lehet.

Kétszeresen láncolt lista

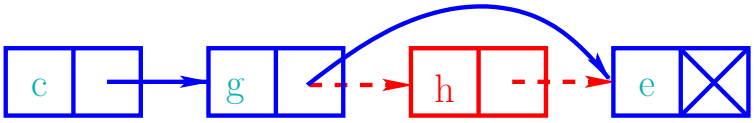


Láncolt lista műveletek

Beszúrás



Törlés



Tömb esetén a beszúrásnál és törlésnél el kell tolni a mögötte levő részt: $c \cdot n$ lépés

Rendezett tömb és láncolt lista

Tehetünk több megszorítást is az egyes adatszekezeteknél. Ilyenkor általában egyes műveleteket gyorsabban, más műveleteket lassabban lehet elvégezni. Az alkalmazástól függ, hogy ez jó-e nekünk.
Ha a tárolandó elemeken adott egy rendezési reláció, akkor megköthetjük, hogy a tömbben vagy a láncolt listában e szerint sorba rendezetten tároljuk az elemeket.

Rendezett tömb

Gyorsabb keserés, lassabb beszúrás, törlés kb. ugyanannyi

Rendezett láncolt lista

Lassabb beszúrás, keresés, törlés kb. ugyanannyi

Rendezési reláció

Definíció

Legyen U egy halmaz, és $<$ egy kétváltozós reláció U-n. Ha $a, b \in U$ és $a < b$, akkor azt mondjuk, hogy a kisebb, mint b.

A $<$ reláció egy (szigorú) rendezés (linear order) , ha teljesülnek a következők:
1. $a \not< a$ minden $a \in U$ elemre ($<$ irreflexív);
2. Ha $a, b, c \in U$, $a < b$, és $b < c$, akkor $a < c$ ($<$ tranzitív);
3. Tetszőleges $a \neq b \in U$ elemekre vagy $a < b$, vagy $b < a$ fennáll ($<$ teljes).
Ha $<$ egy rendezés U-n, akkor az $(U, <)$ párt rendezett halmaznak (ordered set) nevezzük.

Rendezési reláció

Példák:

- $\mathbb{Z}$ az egész számok halmaza. A $<$ rendezés a nagyság szerinti rendezés.
- Az *abc* betűinek halmaza; a $<$ rendezést az *abc-sorrend* adja. Az x betű kisebb, mint az y betű, ha x előbb szerepel az *abc*-sorrendben, mint y.
- Egy *abc* betűiből alkotott szavak halmaza a szótárszerű vagy lexikografikus rendezéssel. $\Rightarrow$ legyen $X = x_1 x_2 \cdots x_k$ és $Y = y_1 y_2 \cdots y_l$ két szó.
  Az X kisebb mint Y, ha vagy $l > k$ és $x_i = y_i$ ha $i = 1, 2, \ldots, k$; vagy pedig $x_j < y_j$ teljesül a legkisebb olyan j indexre, melyre $x_j \neq y_j$. Tehát például *kar* $<$ *karika* és *bor* $<$ *bot*.

Keresés rendezetlen tömbben és láncolt listában

Feladat

Adott az U halmaz véges $S = \{s_1, s_2, \ldots, s_{n-1}, s_n\}$ részhalmaza és $s \in U$.
El akarjuk eldönteni, hogy igaz-e $s \in S$ és ha igen, akkor melyik i-re teljesül $s_i = s$.
Hány összehasonlítás kell?

Itt összehasonlítás: Igaz-e, hogy $s_i = s$?
Válasz: Igen vagy nem.

Legrosszabb esetben minden elemet végig kell nézni $\Longrightarrow$
n összehasonlítás kell legrosszabb esetben
$n/2$ összehasonlítás kell átlagosan
Ez tömb és láncolt lista esetén is így van.

Keresés rendezett tömbben

Bar Kochba játék: gondolok egy számot $1-100$-ig, hány eldöntendő kérdésből lehet kitalálni?

Feladat

Adott az $(U, <)$ rendezett halmaz véges
$S = \{s_1 < s_2 < \ldots < s_{n-1} < s_n\}$ részhalmaza és $s \in U$.
Összehasonlításokkal akarjuk eldönteni, hogy igaz-e $s \in S$ és ha igen, akkor melyik i-re teljesül $s_i = s$.
Hány összehasonlítás kell?

Itt összehasonlítás: Mi a viszonya s-nek és s_i-nek?
Válasz: $s_i \leq s$ vagy $s_i > s$, de a végén szükség lehet egy „rákérdezésre” annak eldöntésére, hogy s benne van-e S-ben.

Lineáris keresés

Sorban mindegyik elemmel összehasonlítjuk.
Költség a legrosszabb esetben: n, mert lehet, hogy pont az utolsó volt.
Költség átlagos esetben esetben: $(n/2) + 1$

Bináris keresés (binary search) rendezett tömbben

Oszd meg és uralkodj: először a középső s_i-vel hasonlítunk.
A válasz kizárja a tömb egyik felét.
A maradék elemek halmaza legyen S_1.
Hasonló feladatot kapunk egy S_1 halmazra, amire viszont $|S_1| \leq |S|/2$.
És így tovább:

$$|S_2| \leq \frac{|S|}{4}, |S_3| \leq \frac{|S|}{2^3}, \ldots |S_k| \leq \frac{|S|}{2^k}$$

Pl. keressük meg, benne van-e 21 az alábbi sorozatban!

$$15, 22, 25, 37, \boxed{48}, 56, 70, 82 \quad (1)$$
$$15, 22, \boxed{25}, 37, 48, 56, 70, 82 \quad (2)$$
$$15, \boxed{22}, 25, 37, 48, 56, 70, 82 \quad (3)$$
$$\boxed{15}, 22, 25, 37, 48, 56, 70, 82 \quad (4)$$

Bináris keresés rendezett tömbben

Addig kell csinálni, amíg $2 > |S_k| \geq 1$ lesz, vagyis $1 \leq \frac{n}{2^k}$.
$\Longrightarrow 2^k \leq n \Longrightarrow k \leq \lfloor \log_2 n \rfloor$
Ekkor még egy „rákérdezés” eldönti, hogy a keresett elem egyenlő-e a megmaradt elemmel. Ez összesen $k+1$ összehasonlítás volt.
$\Longrightarrow k+1 \leq \lfloor \log_2 n \rfloor + 1 = \lceil \log_2(n+1) \rceil$.

Láncolt listában nem érhető el közvetlenül s_i, ezért ezt az algoritmust láncolt listában nem lehet megvalósítani.

Bináris keresés rendezett tömbben

Tétel

Ez optimális, nincs olyan kereső algoritmus, ami minden esetben kevesebb mint $\lceil \log_2(n+1) \rceil$ kérdést használ.

Bizonyítás.

Az ellenség nem is gondol egy számra, csak mindig úgy válaszol, hogy minél többet kelljen kérdezni.
Ha egy kérdést felteszek, és az igen válasz után mondjuk szóba jön x lehetőség, akkor a nem esetén szóba jön még $n-x$ lehetőség.
Az ellenség úgy válaszol, hogy minél több lehetőség maradjon, így el tudja érni, hogy legalább $n/2$ marad. $\Longrightarrow$ k kérdés után is marad még $\frac{n}{2^k}$ lehetőség.
Ha tehát $\frac{n}{2^k} > 1$, akkor nem tudom, hogy az-e a gondolt szám, vagy nincs benne a sorozatban. Tehát még egy kérdésre szükség van.
$\Longrightarrow \frac{n}{2^k} > 1 \Longrightarrow n > 2^k \Longrightarrow \log_2 n > k$. □

Minimumkeresés

Feladat

Adott az $(U, <)$ rendezett halmaz véges $S = \{s_1, s_2, \ldots, s_{n-1}, s_n\}$ részhalmaza.
Összehasonlításokkal keressük meg az S minimális elemét, azaz egy olyan s_i elemet, hogy minden $i \neq j$ esetén $s_i < s_j$.

- Hány összehasonlítás kell a legroszabb esetben?

Minimumkeresés

Tétel

n elem közül a minimális kiválasztásához legroszabb esetben $n - 1$ összehasonlítás kell.

Bizonyítás.

$n - 1$ összehasonlítás mindig elég: Rendezzünk kieséses versenyt, mindig a kisebb elemet megtartva egy-egy összehasonlítás után. Mivel „mindenki pontosan egyszer kap ki a győztest kivéve”, ez $n - 1$ összehasonlítást igényel.
$n - 1$ összehasonlításnál kevesebb nem mindig elég: Erre majd a gráfoknál visszatérünk.

□