Programozás I.

Keresések
Halmazok, halmazműveletek

Sergyán Szabolcs
sergyan.szabolcs@nik.uni-obuda.hu

Óbudai Egyetem
Neumann János Informatikai Kar

November 10, 2013

Tartalom

Tartalom

Keresési algoritmusok

- A feladat egy adott érték (Y) megkeresése egy N elemű sorozatban.
- Nem biztos, hogy a keresett elem benne van a sorozatban.
- Ha benne van a keresett elem a sorozatban, akkor a sorozatbeli indexét kell meghatározni.
- Láttunk már erre megoldást $\rightarrow$ Lineáris keresés

Tartalom

Algoritmus

Eljárás LineárisKeresés(X, N, Y, VAN, $SORSZ$)

  $i \leftarrow 1$

  Ciklus amíg $(i \leq N)$ **és** $(X[i] \neq Y)$

    $i \leftarrow i + 1$

  Ciklus vége

  $VAN \leftarrow (i \leq N)$

  Ha VAN **akkor**

    $SORSZ \leftarrow i$

  Elágazás vége

Eljárás vége

Lineáris keresés

Futási idő

- Minimális lépésszám: 1
- Maximális lépésszám: N (Ezt kapjuk minden olyan esetben is, ha Y nincs benne a sorozatban!)
- Az átlagos futási idő: $O(N)$
- Ezért hívjuk lineáris keresésnek

Ötlet a rekurzív megvalósításhoz

- Tegyük fel, hogy nem rendezett a sorozatunk
- Ha az X sorozat első eleme nem egyezik meg a keresett Y-nal, akkor hívjuk meg ismét a függvényt, de már csak a másodiktól az N-edik elemig terjedő részsorozattal.
- E jelölje a vizsgálandó részsorozat első elemének indexét, U pedig az utolsó elem indexét
- A függvény visszatérési értéke legyen 0, ha Y nincs benne X-ben, egyéb esetben pedig az X-beli indexe annak az elemnek, amely értéke egyenlő Y-nal

Lineáris keresés

Rekurzív megvalósítás

Függvény Keresés(X, E, U, Y)
    Ha $E > U$ **akkor**
        return(0)
    Különben
        Ha $X[E] = Y$ **akkor**
            return(E)
        Különben
            return(Keresés(X, $E + 1$, U, Y))
        Elágazás vége
    Elágazás vége
Függvény vége

Tartalom

1. Keresések
   - Lineáris keresés
   - Keresés rendezett sorozatban
   - Logaritmikus keresés
2. Halmazok, halmazműveletek

Keresés rendezett sorozatban

Vizsgáljuk meg, hogyan lehet módosítani a lineáris keresés algoritmusát, ha a sorozatunk növekvő sorrendben rendezett.

Algoritmus

Eljárás LineárisKeresés(X, N, Y, VAN, $SORSZ$)
  $i \leftarrow 1$
  Ciklus amíg $(i \leq N)$ **és** $(X[i] < Y)$
    $i \leftarrow i + 1$
  Ciklus vége
  $VAN \leftarrow (i \leq N)$ **és** $(X[i] = Y)$
  Ha VAN **akkor**
    $SORSZ \leftarrow i$
  Elágazás vége
Eljárás vége

Futási idő

- Minimális lépésszám: 1 (Akkor is előállhat ez ha Y nincs benne a sorozatban!)
- Maximális lépésszám: N
- Átlagos lépészszám: $\frac{N+1}{2}$ (Nem függ attól, hogy Y benne van-e a sorozatban!)

Nem lehetne ennél jobban kihasználni a sorozat rendezettségét?

Tartalom

1. Keresések
   - Lineáris keresés
   - Keresés rendezett sorozatban
   - Logaritmikus keresés
2. Halmazok, halmazműveletek

Logaritmikus keresés

Alapötlet

- Vizsgáljuk meg először a sorozat középső elemét.
    - Ha a keresett elem (Y) kisebb mint a középső elem, akkor a középső elem előtt szerepelhet a sorozatban.
    - Ha a keresett elem nagyobb mint a középső elem, akkor a középső elem után szerepelhet a sorozatban.
    - Ha a keresett elem megegyezik a középső elemmel, akkor megtaláltuk a sorozatban.
- Folytassuk az eljárást a középső előtti vagy utáni részével a sorozatnak, ha egyáltalán kell folytatni.

Rekurzív megvalósítás

Függvény Keresés(X, E, U, Y)
  Ha $E > U$ **akkor**
    return(0)
  Különben
    $K \leftarrow \left[(E+U)/2 \right]$
    Elágazás
      $X[K] = Y$ **esetén**
        return(K)
      $X[K] < Y$ **esetén**
        return(Keresés(X, $K+1$, U, Y))
      $X[K] > Y$ **esetén**
        return(Keresés(X, E, $K-1$, Y))
    Elágazás vége
  Elágazás vége
Függvény vége

$\left[\cdot\right]$ az egészrész függvényt jelöli

Logaritmikus keresés

Iteratív megvalósítás

Eljárás LogaritmikusKeresés(X, N, Y, VAN, $SORSZ$)
  $E \leftarrow 1$; $U \leftarrow N$
  Ciklus
    $K \leftarrow \left[(E+U)/2\right]$
    Elágazás
      $Y < X[K]$ **esetén**
        $U \leftarrow K - 1$
      $Y > X[K]$ **esetén**
        $E \leftarrow K + 1$
    Elágazás vége
  Amíg $(E \leq U)$ **és** $(X[K] \neq Y)$
  $VAN \leftarrow (E \leq U)$
  Ha VAN **akkor**
    $SORSZ \leftarrow K$
  Elágazás vége
Eljárás vége

$\left[\cdot\right]$ az egészrész függvényt jelöli

Logaritmikus keresés

Futási idő

- Minimális lépésszám: 1 (ha a középső elem az Y)
- Maximális lépésszám: $\lceil 1 + \log_2 N \rceil$
- Átlagos lépésszám: $\lceil \log_2 N \rceil$
- Így már érthető, honnan kapta a nevét az eljárás
- Van, aki felezéses- vagy bináris keresének hívja az eljárást az alapötlet miatt.

$\lceil \cdot \rceil$ a felső egészrész függvényt jelöli

Logaritmikus keresés

Alapötlet alkalmazása

- Ha rendezett a sorozatunk, akkor csak a keresést lehet a felezéses módszerrel javítani?
- Rendezett sorozatok esetén más programozási tételek gyorsítására is használható:
  - Eldöntés
  - Kiválasztás
  - Kiválogatás
  - Megszámlálás

Algoritmus

Eljárás Eldöntés(X, N, Y, VAN)
  $E \leftarrow 1$; $U \leftarrow N$
  Ciklus
    $$K \leftarrow \left[(E + U)/2 \right]$$
    Elágazás
      $Y < X[K]$ **esetén**
        $U \leftarrow K - 1$
      $Y > X[K]$ **esetén**
        $E \leftarrow K + 1$
    Elágazás vége
  Amíg $(E \leq U)$ **és** $(X[K] \neq Y)$
  $VAN \leftarrow (E \leq U)$
Eljárás vége

Kiválasztás

Algoritmus

Eljárás Kiválasztás(X, N, Y, $SORSZ$)
    $E \leftarrow 1$; $U \leftarrow N$
    Ciklus
        $K \leftarrow \left[(E+U)/2\right]$
        Elágazás
            $Y < X[K]$ **esetén**
                $U \leftarrow K - 1$
            $Y > X[K]$ **esetén**
                $E \leftarrow K + 1$
        Elágazás vége
    Amíg $(E \leq U)$ **és** $(X[K] \neq Y)$
    $SORSZ \leftarrow K$
Eljárás vége

Tudjuk, hogy Y biztosan benne van a sorozatban!

Kiválogatás

- Az azonos értékű elemek egymás mellett lesznek.
- Logaritmikus kereséssel megkeresünk egyet ezek közül.
- Ennek ismeretében megkeressük, hogy hol van az első ilyen elem (E) és az utolsó ilyen elem (U).
- Az algoritmus E és U értékét adja vissza.
- A *Megszámlálás* hasonlóan oldható meg: $DB = U - E + 1$.

Kiválogatás

Algoritmus

Eljárás Kiválogatás(X, N, Y, VAN, E, U)
  $E \leftarrow 1$; $U \leftarrow N$
  Ciklus
    $K \leftarrow \left[(E+U)/2\right]$
    Elágazás
      $Y < X[K]$ **esetén**
        $U \leftarrow K - 1$
      $Y > X[K]$ **esetén**
        $E \leftarrow K + 1$
    Elágazás vége
  Amíg $(E \leq U)$ **és** $(X[K] \neq Y)$
  $VAN \leftarrow (E \leq U)$
  Ha VAN **akkor**
    $E \leftarrow K$
    Ciklus amíg $(E > 1)$ **és** $(X[E-1] = Y)$
      $E \leftarrow E - 1$
    Ciklus vége
    $U \leftarrow K$
    Ciklus amíg $(U < N)$ **és** $(X[U+1] = Y)$
      $U \leftarrow U + 1$
    Ciklus vége
  Elágazás vége
Eljárás vége

Tartalom

Halmaz

Definíció

Halmaznak tekintünk egy olyan (növekvő módon) *rendezett* tömböt, melynek minden eleme különböző.

Megvalósítandó eljárások

- Halmaz létrehozása rendezett tömbből (többszörös elemek elhagyása)
- Egy tömbről meghatározni, hogy halmaz-e
- Tartalmazás
- Részhalmaz
- Unió
- Metszet
- Különbség
- Komplementer
- Szimmetrikus differencia

Halmaz létrehozása

Bemenet/kimenet

- Bemenet: N elemű rendezett tömb (X)
- Kimenet: M elemű halmaz (X)

Algoritmus

Eljárás HalmazLétrehozás(X, N, M)
    $j \leftarrow 1$
    Ciklus $i \leftarrow 1$**-től** N**-ig**
        Ha $X[i] \neq X[j]$ **akkor**
            $j \leftarrow j + 1$
            $X[j] \leftarrow X[i]$
        Elágazás vége
    Ciklus vége
    $M \leftarrow j$
Eljárás vége

Halmaz-e?

Bemenet/kimenet

- Bemenet: N (legalább kettő) elemű rendezett tömb: X
- Kimenet: L logikai változó

Algoritmus

Eljárás Halmaz_e(N, X, L)
  $i \leftarrow 2$
  Ciklus amíg $(i \leq N)$ **és** $(X[i] \neq X[i-1])$
    $i \leftarrow i + 1$
  Ciklus vége
  $L \leftarrow (i > N)$
Eljárás vége

Tartalmazás

A *logaritmikus keresés*ből származtatott Eldöntést kell alkalmazni.

Algoritmus

Eljárás Eldöntés(X, N, Y, VAN)
  $E \leftarrow 1$; $U \leftarrow N$
  Ciklus
    $K \leftarrow \left[(E+U)/2 \right]$
    Elágazás
      $Y < X[K]$ **esetén**
        $U \leftarrow K - 1$
      $Y > X[K]$ **esetén**
        $E \leftarrow K + 1$
    Elágazás vége
  Amíg $(E \leq U)$ **és** $(X[K] \neq Y)$
  $VAN \leftarrow (E \leq U)$
Eljárás vége

Matematikai definíció

$A \subseteq B$, ha A minden eleme a B-nek is eleme.

Algoritmus

Eljárás Részhalmaz(X, M, Y, N, L)
  $i \leftarrow 1$; $j \leftarrow 1$
  Ciklus amíg $(i \leq M)$ **és** $(j \leq N)$ **és** $(X[i] \geq Y[j])$
    Ha $X[i] = Y[j]$ **akkor**
      $i \leftarrow i + 1$
    Elágazás vége
    $j \leftarrow j + 1$
  Ciklus vége
  $L \leftarrow (i > M)$
Eljárás vége

Unió

Matematikai defincíció

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

Az Összefuttatás tétel ezt valósítja meg.

Algoritmus

Eljárás Összefuttatás(X, M, Y, N, DB, Z)
  $i \leftarrow 1$; $j \leftarrow 1$; $DB \leftarrow 0$
  $X[M+1] \leftarrow +\infty$; $Y[N+1] \leftarrow +\infty$
  Ciklus amíg $(i < M+1)$ **vagy** $(j < N+1)$
    $DB \leftarrow DB + 1$
    Elágazás
      $X[i] < Y[j]$ **esetén** $Z[DB] \leftarrow X[i]$; $i \leftarrow i + 1$
      $X[i] = Y[j]$ **esetén** $Z[DB] \leftarrow X[i]$; $i \leftarrow i + 1$; $j \leftarrow j + 1$
      $X[i] > Y[j]$ **esetén** $Z[DB] \leftarrow Y[j]$; $j \leftarrow j + 1$
    Elágazás vége
  Ciklus vége
Eljárás vége

Metszet

Matematikai defincíció

$$A \cap B = \{x | x \in A \text{ és } x \in B\}$$

- A korábban tárgyalt Metszet tételben nem használtuk ki, hogy rendezett a sorozat.
- Az Összefuttatás tétel módosításával viszont megoldható a feladat.

Metszet

Algoritmus

Eljárás Metszet(X, M, Y, N, DB, Z)
  $i \leftarrow 1$; $j \leftarrow 1$; $DB \leftarrow 0$
  Ciklus amíg $(i \leq M)$ **és** $(j \leq N)$
    Elágazás
      $X[i] < Y[j]$ **esetén**
        $i \leftarrow i + 1$
      $X[i] > Y[j]$ **esetén**
        $j \leftarrow j + 1$
      $X[i] = Y[j]$ **esetén**
        $DB \leftarrow DB + 1$
        $Z[DB] \leftarrow X[i]$
        $i \leftarrow i + 1$; $j \leftarrow j + 1$
    Elágazás vége
  Ciklus vége
Eljárás vége

Különbség

Matematikai definíció

$$A \setminus B = \{x | x \in A \text{ és } x \notin B\}$$

Algoritmus

Eljárás Különbség(X, M, Y, N, DB, Z)
  $i \leftarrow 1$; $j \leftarrow 1$; $DB \leftarrow 0$
  Ciklus amíg $(i \leq M)$ **és** $(j \leq N)$
    Elágazás
      $X[i] < Y[j]$ **esetén**
        $DB \leftarrow DB + 1$
        $Z[DB] \leftarrow X[i]$
        $i \leftarrow i + 1$
      $X[i] > Y[j]$ **esetén**
        $j \leftarrow j + 1$
      $X[i] = Y[j]$ **esetén**
        $i \leftarrow i + 1$; $j \leftarrow j + 1$
    Elágazás vége
  Ciklus vége
  Ciklus amíg $(i \leq M)$
    $DB \leftarrow DB + 1$; $Z[DB] \leftarrow X[i]$; $i \leftarrow i + 1$
  Ciklus amíg
Eljárás vége

Komplementer

Matematikai definíció

$$\overline{A} = \{x | x \in U \text{ és } x \notin A\},$$

ahol U az univerzális halmaz. Ebből következik, hogy

$$\overline{A} = U \setminus A.$$

Szimmetrikus differencia

Matematikai definíció

$$A \triangle B = (A \setminus B) \cup (B \setminus A)$$

Algoritmus

Eljárás SzimmetrikusDifferencia(X, M, Y, N, DB, Z)
  $i \leftarrow 1$; $j \leftarrow 1$; $DB \leftarrow 0$
  Ciklus amíg ($i \leq M$) **és** ($j \leq N$)
    Elágazás
      $X[i] < Y[j]$ **esetén**
        $DB \leftarrow DB + 1$
        $Z[DB] \leftarrow X[i]$
        $i \leftarrow i + 1$
      $X[i] > Y[j]$ **esetén**
        $DB \leftarrow DB + 1$
        $Z[DB] \leftarrow Y[j]$
        $j \leftarrow j + 1$
      $X[i] = Y[j]$ **esetén**
        $i \leftarrow i + 1$; $j \leftarrow j + 1$
    Elágazás vége
  Ciklus vége
  Ciklus amíg ($i \leq M$)
    $DB \leftarrow DB + 1$; $Z[DB] \leftarrow X[i]$; $i \leftarrow i + 1$
  Ciklus vége
  Ciklus amíg ($j \leq N$)
    $DB \leftarrow DB + 1$; $Z[DB] \leftarrow Y[j]$; $j \leftarrow j + 1$
  Ciklus vége
Eljárás vége

Felhasznált irodalom

- Szlávi Péter, Zsakó László: Módszeres programozás: Programozási tételek (Mikrológia 19). ELTE TTK, 2002