Programozási tételek

Dr. Iványi Péter

Programozási tételek

- A programozási tételek olyan általános algoritmusok, melyekkel programozás során gyakran találkozunk.
- Az algoritmusok általában tömbökkel foglalkoznak, legyen tehát T egy N elemű tömb (1..N)

Összegzés

- Egy tömb elemeinek összegzése
- Könnyen átírható szorzatra vagy más műveletre

```
s:=0
Ciklus i:=1...N
  s:=s+T[i]
Ciklus vége
Ki: s
```

Meszámolás

- Megszámolja, hogy a tömbben hány, adott tulajdonságú elem van
  - Például, negatív számok

```
s:=0
Ciklus i:=1..N
  Ha T[i]<0 akkor s:=s+1
Ciklus vége
Ki: s
```

Eldöntés

- Az algoritmus eldönti, hogy van-e a tömbben adott tulajdonságú elem. Amint talál egyet, a ciklus leáll.
- Ha a ciklus azért állt le, mert túlléptünk a tömb utolsó, vizsgált elemén is, akkor nem volt benne keresett elem

Eldöntés

- Van-e 50 az elemek között

```
i:=1
Ciklus amíg i<=N és T[i]<>50
  i:=i+1
Ciklus vége
Ha i<=N akkor
  ki: "volt 50"
```

Fontos feltételek sorrendje!!

Kiválasztás

- Az algoritmus megadja, hogy a tömbben egy bizonyos elem hol (hányadik helyen) van.
- Csak akkor működik, ha biztosan van ilyen elem

```
i:=1
Ciklus amíg T[i]<>50
  i:=i+1
Ciklus vége
ki: i
```

Keresés

- Az előzőnél biztonságosabb algoritmus: megadja, hogy van-e olyan elem, és ha igen, hányadik.
  (többféle kereső algoritmus van)

```
i:=1
Ciklus amíg i<=N és T[i]<>50
  i:=i+1
Ciklus vége
Ha i<=N akkor
  ki: i
különben
  ki: -1 /* bármilyen más érvénytelen index */
```

Kiválogatás

- Ez az algoritmus egy tömb bizonyos tulajdonságú elemeit teszi egy másik tömbbe.
- db változó számolja, hogy a másik tömbbe hány elem került
- válogassuk ki a negatív számokat.
- Az eredmény a B tömbben lesz
- deklarációnál a B tömböt N eleműre kell választani, hacsak nem tudjuk előre, hány negatív szám van T-ben

Kiválogatás

```
db:=0
Ciklus i:=1..N
  Ha T[i]<0 akkor
    db:=db+1
    B[db]:=T[i]
  Ha vége
Ciklus vége
```

Szétválogatás

- Kiválogatáshoz hasonló, de a nem megfelelő elemeket is tömbbe tesszük

  Szétválogatás két tömbbe

```
dbb:=0
dbc:=0
Ciklus i:=1..N
  Ha T[i]<0 akkor
    dbb:=dbb+1, B[dbb]:=T[i]
  különben
    dbc:=dbc+1, C[dbc]:=T[i]
Ciklus vége
```

Metszet

- két tömb (A[1..N] és B [1..M]) azonos elemeinek kiválogatása C tömbbe
- Az algoritmus lényege: menjünk végik A tömb elemein, és válogassuk ki azokat (kiválogatás), melyek szerepelnek B-ben (eldöntés).
- Visszavezethető a korábbi feladatokra
- C maximális elemszáma N és M közül a kisebbik

Metszet

```
db:=0
Ciklus i:=1..N
  j:=1
  Ciklus amíg j<=M és B[j]<>A[i]
    j:=j+1
  Ciklus vége
  Ha j<=M akkor db:=db+1, C[db]:=A[i]
Ciklus vége
```

Unió

- A és B tömb összes elemét C tömbbe tenni
- Tegyük be C-be A összes elemét, majd B-ből azokat, melyek nem szerepelnek A-ban.
- C elemszáma legfeljebb N+M.

Unió

```
Ciklus i:=1..N
  C[i]:=A[i]
Ciklus vége
db:=N
Ciklus j:=1..M
  i:=1
  Ciklus amíg i<=N és B[j]<>A[i]
    i:=i+1
  Ciklus vége
  Ha i>N akkor db:=db+1, C[db]:=B[j]
Ciklus vége
```

Maximum kiválasztás

- T tömb maximális elemének megkeresése

```
m:=1
Ciklus i:=2..N
  Ha T[i]>T[m] akkor m:=I
Ciklus vége
Ki: m, T[m]
```

- m: a pillanatnyilag talált legnagyobb elem helyét mutatja

Rendezés

- Sokféle van
  - Különböző adatokra
  - Különböző rendezettségre
  - stb

Rendezés maximum kiválasztással

- Az elv: kiválasztjuk a tömb legnagyobb elemét, és berakjuk a tömb végére (vagyis kicseréljük az utolsó elemmel)
- Ezt az eljárást ismételjük a maradék tömbre
- i változó adja meg, hogy hányadik elem fog a helyére kerülni
- A Csere(i,m) eljárás kicseréli a tömb i. és m. elemét

Rendezés maximum kiválasztással

```
Ciklus i:=N..2
  m:=1
  Ciklus j:=2..I
    Ha T[j]>T[m] akkor m:=j
  Ciklus vége
  Csere(i,m)
Ciklus vége
```

Buborékos rendezés

- Végigmegy a tömbön, és ha szomszédos elemeknél rossz a sorrend, megcseréli őket.
- Ez a csere, mint egy buborék, végighalad a tömbön, és a legnagyobb elemet biztosan a tömb végére teszi.
- i változó ismét azt jelzi, hányadik elem kerül a helyére.

Buborékos rendezés

```
Ciklus i:=N..2
  Ciklus j:=1..i-1
    Ha T[J]>T[J+1] akkor Csere(j,j+1)
  Ciklus vége
Ciklus vége
```