Szövegfeldolgozás

ELTE

Szöveges típusok

Karakterábrázolás

- fix kódhossz (ASCII, EBCDIC, ..., több karakterkészlet)
- változó kódhossz (pl. morze, Huffman)
- telex: betű-számváltó
- UTF-8-ban minden ékezetes magyar betű 2 byte-ot foglal el, míg egyéb speciális karakterek akár ennél hosszabbak is lehetnek.

ELTE

Szöveges típusok

Karakterek

Műveletek

- Konverziók (ord, chr)
- relációk
- értékadás
- + (azaz egymásután írás)

ELTE

Szöveges típusok

Karakterhasonlítás

```
kisebb(x,y):
  Konstans B=('A','Á','B',…,'Z')
  x:=nagybetűssé(x)
  y:=nagybetűssé(y)
  xi:=0; yi:=0; i:=1;
  Ciklus amíg (xi=0 vagy yi=0)
    Ha x=B(i) akkor xi:=i
    Ha y=B(i) akkor yí:=i
     i:=i+1
  Ciklus vége
  kisebb:=(xi<yi)
Függvény vége.
```

ELTE

Szöveges típusok

Szövegábrázolás

- fix hossz
- változó hossz
  - karakterszámmal
  - végjellel (pl. 0 kódú karakter)

Műveletek

- hossz(s)
- relációk
- értékadás
- + (azaz egymásután írás)

ELTE

Szöveges típusok

A. A szöveg karakterek tömbje

- indexelés – s[i]

B. A szöveg karakterek sorozata

- első(s), elsőutániak(s), utolsó(s), utolsóelőttiek(s), elejére(s,k), végére(s,k)

C. A szöveg szövegekből áll

- bal(s,db), jobb(s,db), közép(s,k,db) , közép(s,k,v)
- **vagy** részképzés – s[a..b]

ELTE

Szöveges típusok

Szövegfile

- A szövegfile sorok sorozata.
- Minden sor karakterek sorozata, amit sorvég karakter zár le.
- A szövegfile emiatt háromféle-képpen is feldolgozható:
  - soronként
  - karakterenként
  - ’szavanként’

ELTE

Szöveges típusok

Szövegfile ábrázolás

- Nyit(f,Név) assign(f,Név); reset(f)
  assign(f,Név); rewrite(f)
- Zár(f) close(f)
- Olvas(f,kar) read(f,kar)
- SorOlvas(f,sor) readln(f,sor)
- Ír(f,kar) write(f,kar)
- SorÍr(f,sor) writeln(f,sor)
- Sorvége?(f) eoln(f)
- Filevége?(f) eof(f)

ELTE

Szöveges típusok

Szövegfile feldolgozás

```
Feldolgoz(f):
  Nyit(f)
  Ciklus amíg nem Filevége?(f)
    Ciklus amíg nem Sorvége?(f)
      Olvas(f,kar); Ki: kar
    Ciklus vége
    Olvas(f,sorvég)
    Ki: sorvég
  Ciklus vége
  Zár(f)
Eljárás vége.
```

```
while not eof(f) do
begin
 while not eoln(f) do
 begin
  read(f,kar);
  write(kar);
 end;
 readln(f); writeln;
end;
```

ELTE

Szöveges típusok

Szövegfile feldolgozás

```
Feldolgoz(f):
  Nyit(f)
  Ciklus amíg nem Filevége?(f)
    SorOlvas(f,sor); Ki: sor
  Ciklus vége
  Zár(f)
Eljárás vége.
```

```
while not eof(f) do
begin
  readln(f,sor);
  writeln(sor);
end;
```

ELTE

Szöveges típusok általánosítása

Fogalmak (azaz új típusok):

- karakter
- szó
- sor
- lap
- dokumentum

Fogalmak a megjelenítéshez:

- képsor
- képlap

ELTE

Szöveges típusok általánosítása

Műveletek a formázáshoz:

- BalrólLevág
- JobbrólLevág
- Balralgazít
- Jobbralgazít
- Középrelgazít
- Sorkizárt

ELTE

Szöveges típusok általánosítása

Műveletek megvalósítása:

```
BalraIgazít(ks):
  BalrólLevág(ks)
Eljárás vége.
Jobbraigazít(ks):
  JobbrólLevág(ks)
  Ciklus amíg hossz(ks)<KSHOSSZ
    ks:=' '+ks
  Ciklus vége
Eljárás vége.
```

ELTE

Szöveges típusok általánosítása

Műveletek megvalósítása:

```
KözépreIgazít(ks):
  BalrólLevág(ks); JobbrólLevág(ks)
  Ciklus amíg hossz(ks)<KSHOSSZ-1
    ks:=' '+ks+' '
  Ciklus vége
  Ha hossz(ks)<KSHOSSZ
                akkor ks:=ks+' '
Eljárás vége.
```

ELTE

Szöveges típusok általánosítása

Műveletek megvalósítása:

```
Sorkizárt(ks):
 BalrólLevág(ks); JobbrólLevág(ks)
 Szószámlálás(ks,db);
  Szóközszétoszt(ks,db-1,KSHOSSZ)
Eljárás vége.
```

ELTE

Szövegfeldolgozási alapfeladatok

Szűrés: egy szövegből vagy szöveg-file-ból hagyjunk ki bizonyos típusú részeket!

Tömörítés: egy szöveget vagy szövegfile-t alakítsunk át úgy, hogy kevesebb helyet foglaljon (valamint alakítsuk vissza)!

Keresés: egy szövegben vagy szövegfile-ban keressünk egy szöveget!

ELTE

Szövegfeldolgozás: szűrés

Általános feladat: egy szövegből vagy szövegfile-ból hagyjunk ki bizonyos típusú részeket!

- Lehet kiválogatás, ha egyes karakte-reket kell szűrni.
- Lehet kiválogatás előreolvasással, ha karakter-párokat kell szűrni.
- Lehet szókeresés, ha szavakat kell szűrni (de ez is lehet kiválogatás).
- Lehet jelpár keresés, ha páros jelek közötti részt kell szűrni!

ELTE

Szövegfeldolgozás: szűrés

Szóköz szűrés

Példa: Esik az eső ⇒ Esikazeső

```
Szűrés:
  Nyit(f,g)
  Ciklus amíg nem FileVége?(f)
    Olvas(f,kar)
    Ha kar≠' ' akkor Ír(g,kar)
  Ciklus vége
  Zár(f,g)
Eljárás vége.
```

ELTE

Szövegfeldolgozás: szűrés

Karakter párok szűrése (pl. <=) – előreolvasás módszere

```
Szűrés(ks):
  Nyit(f,g); Olvas(f,ekar)
  Ciklus amíg nem FileVége?(f)
    Olvas(f,kar)
    Ha ekar='<' és kar='='
       akkor Olvas(f,ekar)
     különben Ír(g,ekar); ekar:=kar
  Ciklus vége
  Ír(g,ekar); Zár(f,g)
Eljárás vége.
```

ELTE

Szövegfeldolgozás: tömörítés

Általános feladat: egy szöveget alakítsunk át olyan ábrázolásúra, hogy kevesebb helyet foglaljon!

A tömörített szövegnek visszaalakíthatónak kell lenni!

Módszerek:

- karakterek kódolása
- karaktersorozatok kódolása

ELTE

Szövegfeldolgozás: tömörítés

Tömörítés TAB-karakterekkel

TAB karakter jelentése: az aktuális pozíciótól a következő tabulátor-pozícióig szóközöket kell írni!

Tabulációs pozíció (balra igazított):

- fix távolságra egymástól;
- beállítható távolságra egymástól.

Tömörítés: szóközök helyére TAB.

Kicsomagolás: TAB helyére szóközök.

ELTE

Szövegfeldolgozás: tömörítés

Tömörítés TAB-karakterekkel

A bemenő elemek csoportosítása:

- nem szóköz karakter;
- szóközök TAB-pozícióig;
- szóközök nem szóközig.

A kimenő elemek csoportosítása:

- nem szóköz karakter;
- TAB-karakter;
- szóközök nem szóközig.

ELTE

Szövegfeldolgozás: tömörítés

TAB-ok kicsomagolása

A bemenő elemek csoportosítása:

- TAB karakter;
- egyéb karakter.

A kimenő elemek csoportosítása:

- szóközök TAB-pozícióig;
- egyéb karakter.

ELTE

Szövegfeldolgozás: tömörítés

Tömörítés futamhossz kódolással

Futam jelentése: azonos karakterből álló karaktersorozat.

A tömörítés elve: a legalább 4 hosszú futamokról tároljuk a bennük szereplő karaktert, valamint a karakterek darabszámát.

Kicsomagoláshoz tudnunk kell, hogy kódolt értékről van szó, azaz kell egy speciális karakter (pl. Escape).

ELTE

Szövegfeldolgozás: tömörítés

Tömörítés szótárral

Szótár szerepe: a gyakori szavakat egy szótárban tároljuk, majd minden helyen a szótárra hivatkozunk.

Kicsomagoláshoz tudnunk kell, hogy szótári hivatkozásról van szó, azaz kell egy speciális karakter (pl. Escape).

A szótárban csak 256 szó lehet, azaz a szótárbeli sorszámot egyetlen karakterrel adhatjuk meg.

ELTE

Szövegfeldolgozás: keresés

Általános feladat: egy szövegben vagy szövegfile-ban keressünk egy szöveget!

Elemi módszer:

- A keresett szöveg minden karakterét hasonlítsuk a hosszú szöveg elejétől a megfelelő számú karakterrel!
- Ha nem egyezik, akkor a hosszú szövegben 1 karakterrel lépjünk tovább és újra hasonlítsunk!

ELTE

Szövegfeldolgozás: keresés

Elemi módszer

```
Keresés(s,minta,siker,i):
  siker:=Hamis; i:=1
  h:=Hossz(s)-Hossz(minta)+1
  Ciklus amíg i≤h és nem siker
    j:=1
    Ciklus amíg j≤Hossz(minta) és
                  minta(j)=s(i+j-1)
      j:=j+1
    Ciklus vége
    siker:=(j>Hossz(minta))
    Ha nem siker akkor i:=i+1
  Ciklus vége
Eljárás vége.
```

ELTE

Szövegfeldolgozás: keresés

Elemi módszer elemzése

- A külső ciklus lépésszáma: a két szöveg hosszának különbsége. (H)
- A belső ciklus maximális lépésszáma: a keresett szöveg hossza. (K)
- A maximális futási idő: K*H.

Javítási ötletek:

- K csökkentése
- H csökkentése

ELTE

Szövegfeldolgozás: keresés

- Elemi módszer megfordítva
-
```
Keresés(s,minta,siker,i):
  siker:=Hamis; i:=1
  h:=Hossz(s)-Hossz(minta)+1
  Ciklus amíg i≤h és nem siker
    j:=hossz(minta)
    Ciklus amíg j>0 és
                  minta(j)=s(i+j-1)
      j:=j-1
    Ciklus vége
    siker:=(j=0)
    Ha nem siker akkor i:=i+1
  Ciklus vége
Eljárás vége.
```

ELTE

Szövegfeldolgozás: keresés

Eltolás a minta elemei alapján – próbáljuk felhasználni a hasonlítás eredményét a minta több karakterrel eltolására!

- `s:` `abbccacabc`
  ⇳
  `minta:` `abc`
  `eltolás:` `??1`

Az utolsó karakterek nem egyeznek, eltolás jobbra 1 hellyel.

ELTE

Szövegfeldolgozás: keresés

Ha a minta betűi mind különbözőek:

- `s:` `abbccacabc`
  `⇧⇧⇧`
  `minta:` `abc`
  `eltolás:` `?2?`

Hátulról második, eltolás jobbra 2 hellyel.

- `s:` `abbccacabc`
  `⇧⇧⇧`
  `minta:` `abc`
  `eltolás:` `3??`

Hátulról harmadik, eltolás jobbra 3 hellyel.

ELTE

Szövegfeldolgozás: keresés

Ha a minta betűi közel ismétlődnek:

- `s:` `abccbcabbc`
  `⇳⇳`
  `minta:` `abbc`
  `eltolás:` `??21`

Hátulról második, eltolás jobbra 2 hellyel.

- `s:` `abccbcabbc`
  `⇳⇳⇳`
  `minta:` `abbc`
  `eltolás:` `?321`

Hátulról harmadik, eltolás jobbra 3 hellyel.

ELTE

Szövegfeldolgozás: keresés

Ha a minta betűi távol ismétlődnek:

- s: dbbaabba
  ⇧⇧⇧⇧
  minta: abba
  eltolás: 3321

Hátulról negyedik, eltolás csak 3 hellyel.

- s: dbbacabbac
  ⇧⇧⇧⇧⇧
  minta: abbac
  eltolás: 54321

Hátulról ötödik, eltolás mégis 5 hellyel.

ELTE

Szövegfeldolgozás: keresés

Az eltolási vektor

- hátulról monoton növekvő,
- a (hátulról) elsőként előfordulóhoz a hátulról vett sorszáma,
- a nem először előforduló esetén vagy az előzőtől vett távolság; vagy a saját sorszáma (hátulról), ha előző (hátulról) az első ilyen, vagy ha a monotonitási szabály szerinti minimális távolságon túl van vele azonos, akkor a legközelebbi ilyentől vett távolsága.

ELTE

Szövegfeldolgozás: keresés

Eltolás a minta elemei alapján

```
Keresés(s,minta,siker,i):
  siker:=Hamis; i:=1
  h:=Hossz(s)-Hossz(minta)+1
  Ciklus amíg i≤h és nem siker
    j:=hossz(minta)
    Ciklus amíg j>0 és
                  minta(j)=s(i+j-1)
      j:=j-1
    Ciklus vége
    siker:=(j=0)
    Ha nem siker akkor i:=i+ev(j)
  Ciklus vége
Eljárás vége.
```

ELTE

Szövegfeldolgozás: keresés

Boyer-Moore módszer – a szöveg lehetséges karakterei alapján adjuk meg, hogy a mintát milyen messzire kell tolni!

- Ha a minta utáni karakter nem szerepel a mintában, akkor a mintát e mögé kell eltolni!
- Ha szerepel, akkor a mintabeli utolsó előfordulását kell ide tolni!

ELTE

Szövegfeldolgozás: keresés

Rabin-Karp módszer – a szöveget ne karakterenként hasonlítsuk!

- Tekintsük a keresendő, m hosszúságú mintát úgy, mint egy d alapú számrendszerben felírt egész számot, ahol d=a szövegben előfordulható jelek száma!
- A mintából és a szöveg részeiből is egy-egy ilyen számot készítünk, majd számonként hasonlítunk.

ELTE

Szövegfeldolgozás: keresés

Rabin-Karp módszer

- k(i):=Kód(minta(i)); z(i):=Kód(s(i))
- $X:=k(1)*d^{m-1}+k(2)*d^{m-2}+...+k(m-1)*d+k(m)$
- $Y(1):=z(1)*d^{m-1}+z(2)*d^{m-2}+...+z(m)$
- $Y(i):=(Y(i-1) \bmod d^{m-1})*d+z(i+m-1)$

Probléma: X és Y(i) túl nagy lehet.

Megoldás: X mod P és Y(i) mod P használata. Ekkor ha X=Y(i), abból még nem biztos, hogy szövegegyezés van, azt még külön meg kell nézni!


Vége