

8. Sortimine (S)**8.1. Üldist**

S - andmekogumi järjestamine mingi tunnuse järgi

25% arvutiajast sortimisele ja otsimisele

palju algoritme

vaatleme massiivi S

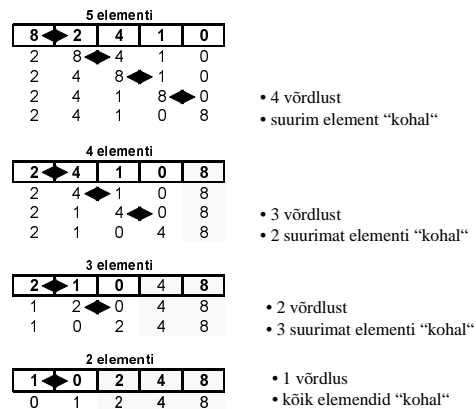
Kirjandus:

1. Jürgenson, R. Programmitehnika II. Tln., EMI, mapp 1-74, 1993.

2. Jürgenson, R. Sortimine ja otsimine. Tln., TTÜ, 1997.

8.2. MullS

Massiiv käiakse korduvalt läbi ja võrreldakse 2 naaberelementi, kui nad on "vales järjekorras", siis nad vahetatakse

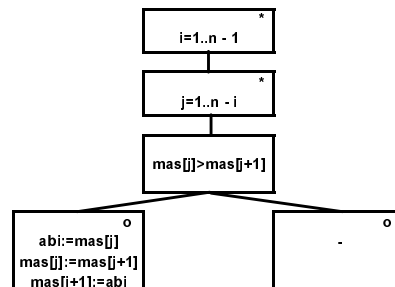


Kokkuvõte:

1. Massiiv tuleb läbi käia n-1 (4) korda
2. 1. kord on n-1 (4) võrdlust, 1. korra lõpus on suurim element "kohal",
 2. kord on n-2 (3) võrdlust, 2. korra lõpus on 2 suurimat elementi "kohal", jne., viimane kord on 1 võrdlus, viimase korra lõpus on kõik elemendid "kohal", s.t. massiiv on sortitud

| | | |
|-------------------------|-----|--------|
| n - läbikäimiste arv | i | j |
| i - läbikäimiste indeks | 1 | 1..4 |
| j - 1. naabri indeks | 2 | 1..3 |
| | ... | |
| | n-1 | 1(..1) |

algoritm



miks *mulli* algoritm?

algoritmi võimalik teha paremaks!

9. Failid (F)**9.1. Üldist**

- F - struktuurne andmetüüp
- ühte ja sama tüüpi komponentide jada, F-i pikkus ei ole määratud

| | | | | | |
|----|----|----|----|-----|----|
| 1. | 2. | 3. | 4. | ... | n. |
|----|----|----|----|-----|----|

- väga suurte andmekogumite hoidmiseks välismälus
- säilib pärast prog. töö lõppu; F-e saab kasutada mitmes programmis
- ainult 2 tegevust:
 - 1) F läbivaatus (järjestikku alates 1. -st); komp.-e vahele jätta, uuendada ega kustutada ei saa; ei saa pöörduda suvalise komponendi poole; PASCAL-li failid - jadafailid; F-i uuendamiseks tuleb moodustada uus F
 - 2) F loomine (moodustamine) - uute komp.-de lisamine F lõppu
- mistahes ajahetkel pääseme kallale ainult 1-le komp.-le
- sisemised ja välised (nimi programmi päises);
- standardfailid (input,output)

9.1. Kirjeldamine

type fail = *file of* tüüp; ← failikomponendi tüüp (mistahes tüüp)

var f: fail;
var f: *file of* tüüp;

F-l saab olla 1 kindel olek:

- läbivaatuse (lugemise)
- loomise (kirjutamise)

Enne faili töötlemist tuleb F avada käsuga

open (f, 'parim.dat', p); p - F-i olek
= 'new', kui loome uue faili
= 'old', kui avame olemasoleva

- f - loogiline nimi
parim.dat - füüsiline nimi (kettal)
- F-i olek määratakse käskudega:
- reset(f);** - läbivaatamise olek
 - rewrite(f);** - loomise olek (NB! See teeb F tühjaks)

F-ga seotud F lõpu tunnus;
selle tuvastamiseks kasutame funktsiooni **eof(f)**

F-i moodustamine:

```
rewrite(f);
while vajakirjutada do
begin
    moodusta komponent;
    write(f, komponent)
end;
```

F-ist lugemine:

```
reset(f);
while not eof(f) do
begin
    read(f, komponent);
    töötle komponent'i
end;
```

Failide ja massiivide võrdlus

| | |
|---|---|
| F-i pikkus määrata | massiivil lõplik |
| F-i elemente saab töödelda ainult järjestikku | massiivil suvalises järjekorras (vastavalt indeksile) |

9.3. Tekstifailid

- failid, mille komponendi tüüp on **char**, on Pascal'is eritähendusega need on nn. tekstifailid (tf)

Seega: *file of char = text*
(packed file of char)

on tf - d

tf koosneb ridadest, mis on üksteisest eraldatud spetsiaalsete juhtmärkidega (rea lõpp); igas reas 0 või rohkem märki

tf puhul kasutatakse lisaks mistahes F-de puhul kasutatavatele

read, write ja **eof**-le veel :

readln, writeln, coln, page

input ja output on standardfailid, s.t. tegelikult on nad standartsete nimedega tekstifailid (*input,output:text*;))

Neid faile ei ole vaja kirjeldada ja ei tohi

```
reset(input)
rewrite(output)
```

Kuna need 2 nime on standartsed, siis võib nad ära jätta:

```
read(x) = read(input,x)
write(x)= write(output,x)
eof    = eof(input)
coln   = coln(input)
```

kui write ilma faili nimeta, siis loetakse selleks output
kui read ilma faili nimeta, siis loetakse selleks input

writeln(f) lisab rea lõpu tunnuse

F-i väljastamisel F jagatakse ridadeks

page(f) seab väljundil ette uue lk alguse

F-i väljastamisel F jagatakse lk-deks

9.4. Kirjete F moodustamise protseduur (näide)

Ülesanne:

Moodusta õpilaste F, kus iga õpilase kohta on järgmiste väljadega kirje :

- eesnimi (string)
- perenimi (string)
- klass (string)
- edukus (5-elementiline vektor väärtustega 0..5)

Kui ei ole ette antud stringide pikkusi, siis tuleks stringide pikkused valida nii, et väljastamisel mahuks kirje ära 1 ritta, s. t. F või massiiv oleks väljastatav tabelina

antud juhul valime nimede pikkuseks 15 märki ja klassi tähistuse pikkuseks 4 märki

seega peaks meil peaprogrammis olema kirjeldatud:

```
type
  vektor = array [1..5] of 0..5;
  opilane=record
    eesnimi: varying[15];
    perenimi: varying[15];
    klass:   varying[4];
    edukus:  vektor
  end;
  opilased=array [1..10] of opilane;
  opfail = file of opilane;

var
  mas:  opilased; {massiiv}
  kf:   opfail;   {kirjete fail}
  tf:   text;     {teksti fail}
```

tingimus: lähteandmed olgu tekstiF-s *lahte.txt*

Soovitav lähteteksti F moodustada nii, et iga väli eraldi real

Jaan
 Tamm
 12B
 3 4 5 4 3
 Sirje
 Kuusk
 8A
 5 4 5 4 3

...

sel juhul on tekstiF-st lugemise programm lihtne

Soovitatav kõik F-de avamised ja sulgemised teha peaprogrammis

Kui töö F-ga on lõpetatud, siis on soovitatav F sulgeda

F suletakse protseduuriga

close (f) f - loogiline nimi

antud juhul peaprogrammis - soovitatav kohe peaprogrammi algul

open (tf, 'lahte.txt', 'old');

open (kf, 'lahte.dat', 'new');

...

ja peaprogrammi lõpus

close(tf);

close(kf);