

Töö nr.2 - paiskadresseerimine
Erki Suurjaak
970772
LAP31

Sisendandmed olid vastavalt matriklinumbrile järgmised :

<sonad.scr>

Youngsters

years

old

have

used

computer

more

than

other

age

groups

Three

quarters

of

them

have

used

it

when

only

every

fifth

person

in

their

parents'

generation

and

every

twentieth

person

in

their

grandparents'

generation

have

done

that

Look
Graph
</sonad.src>

Nendest võttis programm uusi sõnu niikaua, kuni erinevaid sõna oli kokku 28.
Matriklinumbri järgi oli 3. variandi samm 35.

1. Arvutuste ja paigutuste skeemid

1.1 Samm = 1

```
1) 23>Youngsters
2) 13>years
3) 6>old
4) 0>have
5) 12>used
6) 3>computer
7) 17>more
8) 5>than
9) 14>other
10) 3> 4>age
11) 21>groups
12) 5> 6> 7>Three
13) 7> 8>quarters
14) 0> 1>of
15) 5> 6> 7> 8> 9>them
16) 0>have
17) 12>used
18) 7> 8> 9>10>it
19) 24>when
20) 8> 9>10>11>only
21) 2>every
22) 16>fifth
23) 26>person
24) 1> 2> 3> 4> 5> 6> 7> 8> 9>10>11>12>13>14>15>in
25) 5> 6> 7> 8> 9>10>11>12>13>14>15>16>17>18>their
26) 22>parents'
27) 8> 9>10>11>12>13>14>15>16>17>18>19>generation
28) 10>11>12>13>14>15>16>17>18>19>20>and
29) 2>every
30) 20>21>22>23>24>25>twentieth
31) 26>person
32) 1> 2> 3> 4> 5> 6> 7> 8> 9>10>11>12>13>14>15>in
33) 5> 6> 7> 8> 9>10>11>12>13>14>15>16>17>18>their
34) 21>22>23>24>25>26>27>grandparents'
```

Kollisioone oli 15

Võrdlusi oli 101

1.2 Samm = 35 (vastavalt matriklinumbrile)

- 1) 23>Youngsters
- 2) 13>years
- 3) 6>old
- 4) 0>have
- 5) 12>used
- 6) 3>computer
- 7) 17>more
- 8) 5>than
- 9) 14>other
- 10) 3> 7>age
- 11) 21>groups
- 12) 5> 9>Three
- 13) 7>11>quarters
- 14) 0> 4>of
- 15) 5> 9>13>17>21>25>them
- 16) 0>have
- 17) 12>used
- 18) 7>11>15>it
- 19) 24>when
- 20) 8>only
- 21) 2>every
- 22) 16>fifth
- 23) 26>person
- 24) 1>in
- 25) 5> 9>13>17>21>25>29>their
- 26) 22>parents'
- 27) 8>12>16>20>generation
- 28) 10>and
- 29) 2>every
- 30) 20>24>28>twentieth
- 31) 26>person
- 32) 1>in
- 33) 5> 9>13>17>21>25>29>their
- 34) 21>25>29> 2> 6>10>14>18>grandparents'

Kollisioone oli 11

Võrdlusi oli 35

1.3 Samm = (h mod (maht - 2)) + 1

1) 23>Youngsters
2) 13>years
3) 6>old
4) 0>have
5) 12>used
6) 3>computer
7) 17>more
8) 5>than
9) 14>other
10) 3>10>age
11) 21>groups
12) 5>23>10>28>Three
13) 7>quarters
14) 0>21>11>of
15) 5>23>10>28>15>them
16) 0>have
17) 12>used
18) 7>19>it
19) 24>when
20) 8>only
21) 2>every
22) 16>fifth
23) 26>person
24) 1>in
25) 5>23>10>28>15> 2>20>their
26) 22>parents'
27) 8> 5> 2>30>generation
28) 10>24> 7>21> 4>and
29) 2>every
30) 20>24>28> 1> 5> 9>twentieth
31) 26>person
32) 1>in
33) 5>23>10>28>15> 2>20>their
34) 21> 2>14>26> 7>19> 0>12>24> 5>17>29>grandparents'

Kollisioone oli 11

Võrdlusi oli 46

1.4 Kasutatakse kolmeelemendilisi pakette.

Samm = 1

- 1) 4.0>Youngsters
- 2) 1.0>years
- 3) 9.0>old
- 4) 6.0>have
- 5) 2.0>used
- 6) 10.0>computer
- 7) 3.0>more
- 8) 1.0>1.1>than
- 9) 1.0>1.1>1.2>other
- 10) 0.0>age
- 11) 3.0>3.1>groups
- 12) 1.0>1.1>1.2>2.0>2.1>Three
- 13) 8.0>quarters
- 14) 7.0>of
- 15) 1.0>1.1>1.2>2.0>2.1>2.2>them
- 16) 6.0>have
- 17) 2.0>used
- 18) 2.0>2.1>2.2>3.0>3.1>3.2>it
- 19) 6.0>6.1>when
- 20) 10.0>10.1>only
- 21) 8.0>8.1>every
- 22) 2.0>2.1>2.2>3.0>3.1>3.2>4.0>4.1>fifth
- 23) 5.0>person
- 24) 0.0>0.1>in
- 25) 1.0>1.1>1.2>2.0>2.1>2.2>3.0>3.1>3.2>4.0>4.1>4.2>their
- 26) 4.0>4.1>4.2>5.0>5.1>parents'
- 27) 9.0>9.1>generation
- 28) 2.0>2.1>2.2>3.0>3.1>3.2>4.0>4.1>4.2>5.0>5.1>5.2>and
- 29) 8.0>8.1>every
- 30) 6.0>6.1>6.2>twentieth
- 31) 5.0>person
- 32) 0.0>0.1>in
- 33) 1.0>1.1>1.2>2.0>2.1>2.2>3.0>3.1>3.2>4.0>4.1>4.2>their
- 34) 3.0>3.1>3.2>4.0>4.1>4.2>5.0>5.1>5.2>6.0>6.1>6.2>7.0>7.1>grandparents'

Kollisioone oli 20

Võrdlusi oli 84

2. Hinnangud

2.1 Võrdluste arvu poolest on efektiivsus järgmine :

101 - samm = 1

35 - samm = 35

46 - samm = $(h \bmod (\text{maht} - 2)) + 1$

84 - kasutatakse 3-elemendilisi pakette sammuga 1

Ehk efektiivsuse järjekorras :

35 - samm = 35

46 - samm = $(h \bmod (\text{maht} - 2)) + 1$

84 - kasutatakse 3-elemendilisi pakette sammuga 1

101 - samm = 1

2.2 Kollisioonide põhjal hinnates :

15 - samm = 1

11 - samm = 35

11 - samm = $(h \bmod (\text{maht} - 2)) + 1$

20 - kasutatakse 3-elemendilisi pakette sammuga 1

Ehk efektiivsuse järjekorras :

11 - samm = 35

11 - samm = $(h \bmod (\text{maht} - 2)) + 1$

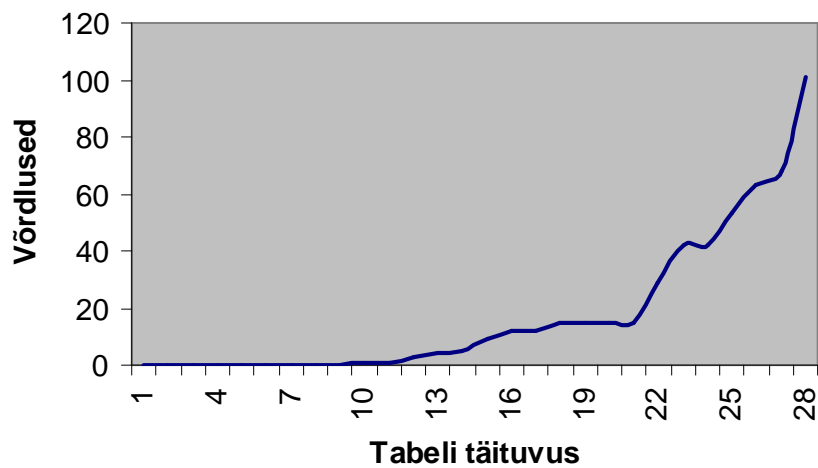
15 - samm = 1

20 - kasutatakse 3-elemendilisi pakette sammuga 1

Mõlema väärtuse põhjal hinnates selgub, et optimaalne on variant, kus samm = 35.

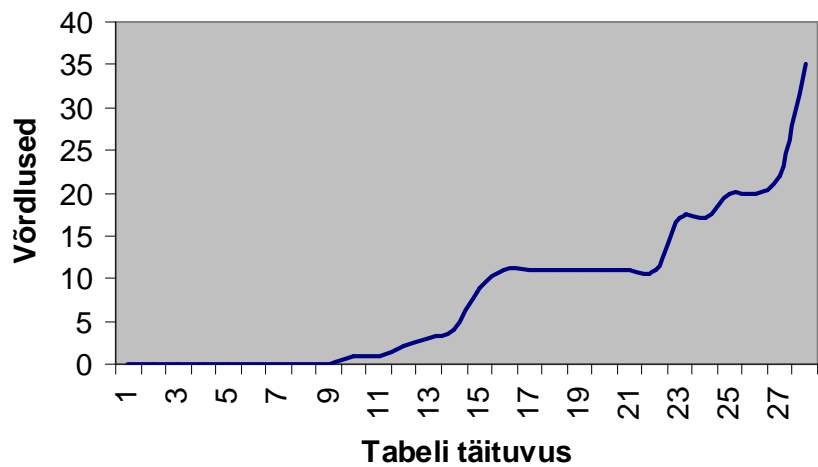
3. Võrdluste sõltuvuse tabeli täituvusastmest

3.1 Samm = 1



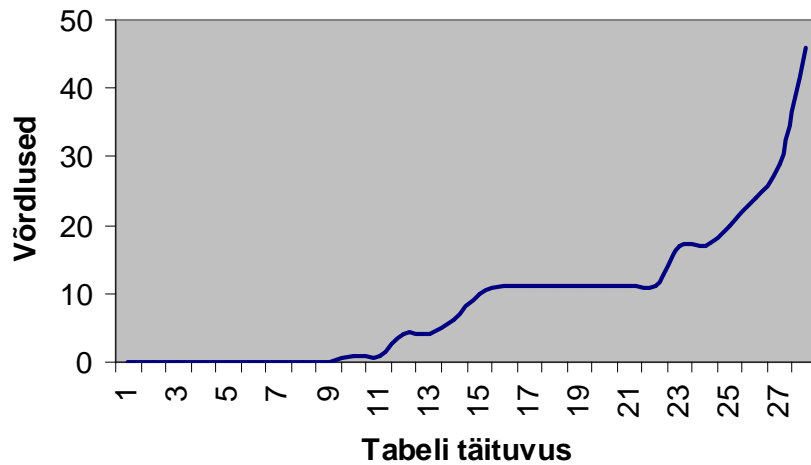
Võrdluste arv kasvab ühtlaselt ning tabeli täitudes kasv kiireneb märgatavalt.

3.2 Samm = 35



Võrdluste arv ei kasva eriti kiiresti ja eriti palju.

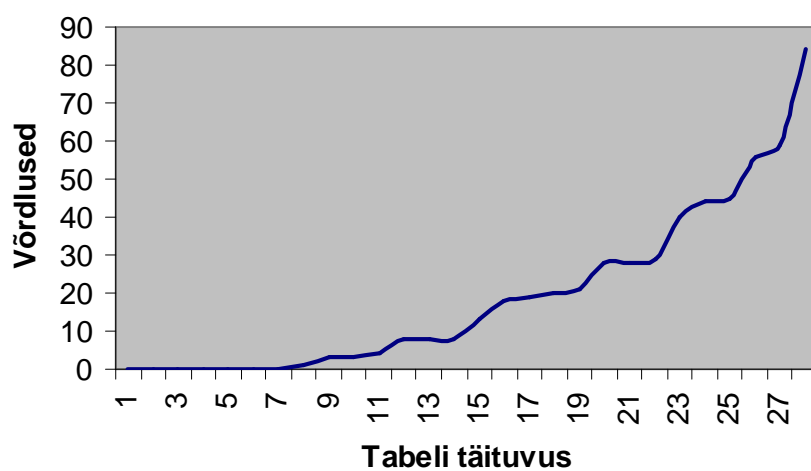
$$3.3 \text{ Samm} = (h \bmod (\text{maht} - 2)) + 1$$



Võrdluste arv kasvab alguses vähe, umbes 70% täidetuse pealt hakkab kiiresti suurenema.

$$3.4 \text{ Samm} = 1$$

Kasutatakse 3-elementilisi pakette



Võrdluste arv kasvab enam-vähem stabiilselt ja võrreldes teistega väga kiirelt.

4. Järeldus

Kõige efektiivsemaks osutus variant, kus samm = 35.

5. Programmi lähtekood

```
PROGRAM PaiskFunktsioon;

Const maht = 31; {Paisktabeli maht, kui ei kasutata 3-elemendilisi pakette }
      maxsonu = 28; {Maksimaalne paisatavate sõnade arv}

      {Siit leitakse vastavalt symbolile vastav number}
      kooditabel : array['A'..'Z'] of byte = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
      12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26);

var paisktabel : array[0..maht + 1] of string[13]; {Viimast kahte elementi
kasutatakse                                     3-elemendiste pakettide
juures}
      i, j,                                {Loendurid}
      samm,
      paiskakuhu,                          {Sõna aadress paisktabelis. 3-elemendiste pakettide}
      {puhul paketi number}
      koht : byte;                         {3-elemendiste pakettide korral paketi elemendi nr}
      kollisioonidearv, v6rdlustearv : word;
      sona : string[13];                  {Parajasti paisatav sõna}
      e,                                  {Siit loetakse sisse paisatavad laused}
      f,                                  {Siia kirjutatakse tabeliseisud}
      g : text;                           {Siia kirjutatakse v6rdluste arv ja tabeli t2ituvus}
      {graafiku jaoks}

{Annab tulemuseks suurte t,htedega stringi}
FUNCTION shift (shiftitav : string) : string;
var s1 : byte;

begin
for s1 := 1 to length(shiftitav) do shiftitav[s1] := upcase (shiftitav[s1]);
shift := shiftitav;
end;

FUNCTION LeiaKood (leitavt2ht : char) : byte;
      {Leitakse t2he kood kooditabeli j2rgi}
begin
if not (leitavt2ht = ' ') then leiakood := kooditabel[upcase(leitavt2ht)]
else leiakood := 0;
end;

FUNCTION Aadress (asi : string) : byte;
      {Leitakse sõna aadress esimese ja teise t2he j2rgi}
begin
aadress := leiakood(asi[1])*27 + leiakood(asi[2]) mod maht;
end;

BEGIN

Assign (e, 'sonad.src');
Assign (f, 'valjund');
Assign (g, 'graafik.dat');
Reset (e);
Rewrite (f);
Rewrite (g);

{----- Samm = 1 -----}
```

```

writeln(f, 'Samm = 1');
writeln (f);
writeln (g, 'Samm = 1');
kollisioonidearv := 0;
v6rdlustearv := 0;

for i := maht - 1 downto 0 do paisktabel[i] := '';
samm := 1;
j := 0;

repeat
inc (i);
write (f, i:2,') ');
readln (e, sona);
paiskakuhu := aadress(sona) mod maht;
write (f, paiskakuhu:2, '>');

while (paisktabel[paiskakuhu] > '')
and (shift(paisktabel[paiskakuhu]) <> shift(sona)) do
begin
inc (v6rdlustearv);
paiskakuhu := (paiskakuhu + samm) mod maht;
write (f, paiskakuhu:2, '>');
end;

if (shift(paisktabel[paiskakuhu]) <> shift(sona)) then inc(j);
if paiskakuhu <> aadress (sona) mod maht then inc (kollisioonidearv);
paisktabel[paiskakuhu] := sona;
writeln (f, sona);
writeln (g, i:8, v6rdlustearv:8);
until j = maxsonu;

writeln (f);
writeln (f, 'Kollisioone oli ', kollisioonidearv);
writeln (f, 'Värdlusi oli ', v6rdlustearv);
writeln (f, '_____');
writeln (g, '_____');
writeln (f);

{----- Samm = 35 -----}

writeln(f, 'Samm = 35 (vastavalt matriklinumbrile)');
writeln (f);
writeln (g, 'Samm = 35');
kollisioonidearv := 0;
v6rdlustearv := 0;
for i := maht - 1 downto 0 do paisktabel[i] := '';
Reset (e);
j := 0;
samm := 35;
repeat
inc (i);
write (f, i:2,') ');
readln (e, sona);
paiskakuhu := aadress(sona) mod maht;
write (f, paiskakuhu:2, '>');

while (paisktabel[paiskakuhu] > '')
and (shift(paisktabel[paiskakuhu]) <> shift(sona)) do
begin
inc (v6rdlustearv);
paiskakuhu := (paiskakuhu + samm) mod maht;
write (f, paiskakuhu:2, '>');
end;

if (shift(paisktabel[paiskakuhu]) <> shift(sona)) then inc(j);
if paiskakuhu <> aadress (sona) mod maht then inc (kollisioonidearv);
paisktabel[paiskakuhu] := sona;
writeln (f, sona);

```

```

writeln (g, i:8, v6rdlustearv:8);
until j = maxsonu;

writeln (f);
writeln (f, 'Kollisioone oli ', kollisioonidearv);
writeln (f, 'Värdlusi oli ', v6rdlustearv);
writeln (f, '_____');
writeln (g, '_____');
writeln (f);

{----- Samm = (h mod (maht - 2)) + 1 -----}

writeln(f, 'Samm = (h mod (maht - 2)) + 1');
writeln (f);
writeln (g, 'Samm = (h mod (maht - 2)) + 1');
kollisioonidearv := 0;
v6rdlustearv := 0;
for i := maht - 1 downto 0 do paisktabel[i] := '';
Reset (e);
j := 0;
repeat
  inc (i);
  write (f, i:2,') ');
  readln (e, sona);
  paiskakuhu := aadress(sona) mod maht;
  write (f, paiskakuhu:2, '>');

  while (paisktabel[paiskakuhu] > '')
    and (shift(paisktabel[paiskakuhu]) <> shift(sona)) do
    begin
      inc (v6rdlustearv);
      samm := ((leiakood(sona[1])*maxsonu + leiakood(sona[2]))
        mod (maht - 2)) + 1;
      paiskakuhu := (paiskakuhu + samm) mod maht;
      write (f, paiskakuhu:2, '>');
    end;

  if (shift(paisktabel[paiskakuhu]) <> shift(sona)) then inc(j);
  if paiskakuhu <> aadress (sona) mod maht then inc (kollisioonidearv);
  paisktabel[paiskakuhu] := sona;
  writeln (f, sona);
  writeln (g, i:8, v6rdlustearv:8);
until j = maxsonu;

writeln (f);
writeln (f, 'Kollisioone oli ', kollisioonidearv);
writeln (f, 'Värdlusi oli ', v6rdlustearv);
writeln (f, '_____');
writeln (g, '_____');
writeln (f);

{----- Kasutatakse 3-elemendilisi pakette, samm = 1 -----}

writeln (f, 'Kasutatakse kolmeelemendilisi pakette. ');
writeln(f, 'Samm = 1');
writeln (g, 'Kasutatakse kolmeelemendilisi pakette. ');
writeln (g, 'Samm = 1');
writeln (f);
kollisioonidearv := 0;
v6rdlustearv := 0;
for i := maht - 1 downto 0 do paisktabel[i] := '';
Reset (e);

samm := 1;
j := 0;
repeat
  inc (i);
  koht := 0;

```

```

write (f, i:2,') ');
readln (e, sona);
paiskakuhu := (aadress (sona) div 3 mod 11);
write (f, paiskakuhu, '.', koht, '>');

while (paisktabel[3*paiskakuhu + koht] > '')
and (shift(paisktabel[3*paiskakuhu + koht]) <> shift(sona))
do
begin
inc (v6rdlustearv);
koht := (koht + 1) mod 3;
if koht = 0 then paiskakuhu := (paiskakuhu + samm) mod 11;
write (f, paiskakuhu, '.', koht, '>');
end;

if (shift(paisktabel[3*paiskakuhu + koht]) <> shift(sona)) then inc(j);
if (koht > 0) or (paiskakuhu <> aadress (sona) div 3 mod 11)
then inc (kollisioonidearv);
paisktabel[3*paiskakuhu + koht] := sona;
writeln (f, sona);
writeln (g, i:8, v6rdlustearv:8);
until j = maxsonu;

writeln (f);
writeln (f, 'Kollisioone oli ', kollisioonidearv);
writeln (f, 'Värdlusi oli ', v6rdlustearv);
writeln (f, '_____');
writeln (g, '_____');
writeln (f);
Close (e);
Close (f);
Close (g);
END.

```