Olgu antud massiiv $a_1, \ldots, a_n$. Kui mitme võrdlemisega on sealt võimalik leida minimaalne element?

```
m := 1
for i := 2 to n do
    if a_i < a_m then
        m := i
return m
```

$n - 1$ võrdlust. Kas vähemaga saab?

Tuleb leida minimaalne element n-st olemasolevast. Meil on n võimalikku vastusevarianti, seega ei saa küsimusi olla vähem kui $\lceil \log n \rceil$.

$\lceil \log n \rceil$ kas-küsimusega on tõepoolest võimalik vähim element leida:

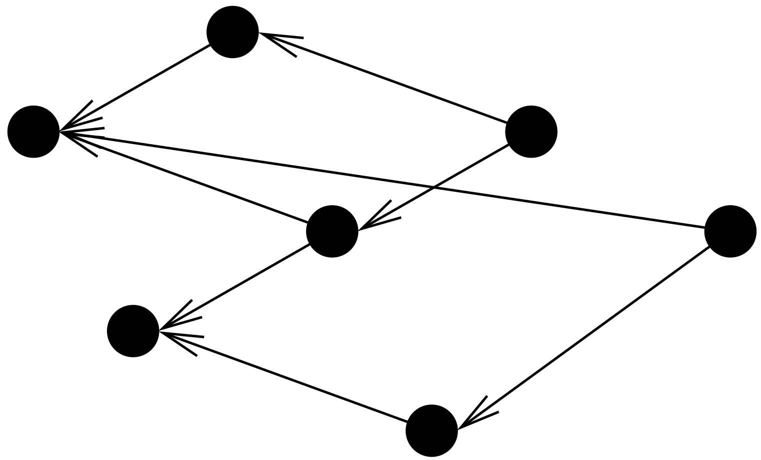
1. Kas vähim element on massiivi esimeses pooles?
2. Kas vähim element on

   jah: massiivi esimeses veerandis?

   ei: massiivi kolmandas veerandis?
3. kaheksandik, kuueteistkümnendik, jne.

Aga need küsimused ei ole kujul „$a_i < a_j$?“

Kui me oleme elementidest $a_1, \ldots, a_n$ mingeid paare omavahel võrrelnud, siis vähimaks elemendiks võib olla iga element, mille jaoks me pole leidnud mõnda elementi, mis temast veel väiksem oleks.



Igal võrdlusel suureneb elementide arv, millest me väiksemat elementi teame, ülimalt ühe võrra.

Seega on tarvis vähemalt $n-1$ võrdlust.

Kui mitut võrdlemist on vaja, et leida massiivi $a_1, \ldots, a_n$ vähim ja suurim element?

$2n - 2$ on piisav (vähima jaoks $n - 1$ ja suurima jaoks ka $n - 1$).

Loeme, et massiivis on paarisarv elemente

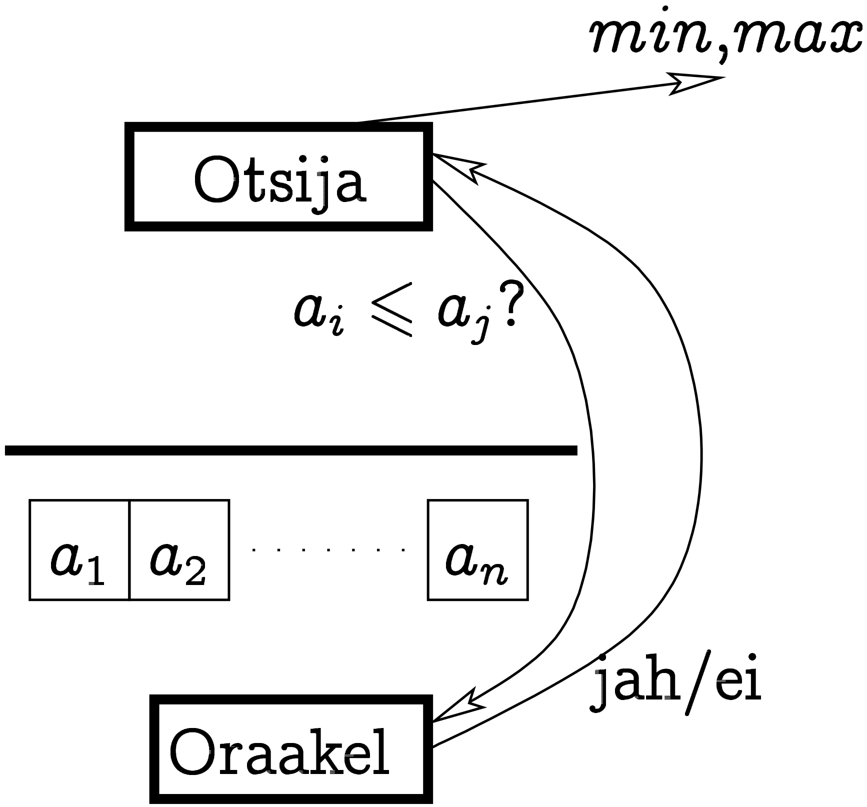
if $a_1 \leqslant a_2$ **then**
    $min := 1; max := 2$
else
    $min := 2; max := 1$
$\ldots$

```
for i := 2 to n/2
    if a_{2i-1} ⩽ a_{2i} then
        if a_{2i-1} < a_{min} then min := 2i - 1
        if a_{2i} > a_{max} then max := 2i
    else
        if a_{2i} < a_{min} then min := 2i
        if a_{2i-1} > a_{max} then max := 2i - 1
```

Kokku $3n/2 - 2$ võrdlust.

Kui oleks $(n + 1)$ elementi (paaritu arv), siis oleks $3n/2$ võrdlust. Seega üldjuhul $\lceil 3n/2 \rceil - 2$ võrdlust.

Kas vähemaga saab?



Olgu $a_1, \ldots, a_n$ Otsija töö alguses fikseerimata. Oraakel fikseerib nad (õigemini nende järjekorra) töö käigus, eesmärgiga anda Otsijale võimalikult palju tööd.

Oraakel ei tohi töö käigus vastuolulisi vastuseid anda. Näiteks $a_1 < a_2$, $a_2 < a_3$ ja $a_3 < a_1$.

Töö käigus on oraakel fikseerinud elementide $a_1, \ldots, a_n$ järjekorra, või on vähemalt pannud sellele järjekorrale mingeid kitsendusi.

Kuna oraakel ei andnud vastuolulisi vastuseid, siis leidub mingi massiiv, kus elementide järjekord neid kitsendusi rahuldab.

Kui otsija töötab selle massiiviga, siis läheb tal sama palju aega, kui oraakliga suheldes läks.

Töö alguses on n võimalikku vähimat elementi ja n võimalikku suurimat elementi.

Kui mõne elemendi jaoks on leitud temast väiksem/suurem element, siis see element ei saa enam vähim/suurim olla.

Töö lõpuks peab järgi jääma ainult 1 võimalik vähim element ja 1 võimalik suurim element.

Oraakel suudab tagada, et see ei juhtuks varem, kui $\lceil 3n/2 \rceil - 2$ päringut on esitatud.

Ta peab pidama arvet, millised elemendid on veel võimalikud vähimad või suurimad.

Kui oraaklilt küsitakse „$a_i \leqslant a_j$?“, siis sõltuvalt sellest, kas i ja j on võimalikud vähimad või suurimad elemendid, tuleb vastata järgmiselt:

$i \backslash j$	vs	v	s	
vs	X	ei	jah	X
v	jah	X	jah	jah
s	ei	ei	X	ei
	X	ei	jah	?

$i \backslash j$	vs	v	s	
vs	2	1	1	1
v	1	1	0	0
s	1	0	1	0
	1	0	0	0

„X“ tähendab, et võib vastata mõlemat moodi.
„?“ tähendab, et vastata tuleb kooskõlas eelmiste vastustega.

Parempoolne tabel näitab võimalike vähimate ja võimalike suurimate elementide arvu vähenemist vastates.

Võimalike vähimate ja suurimate elementide arv väheneb töö käigus $2n - 2$ võrra.

Ühel võrdlemisel väheneb ta ülimalt 2 võrra.

2 võrra väheneb ta ainult siis, kui mõlemad võrreldavad elemendid on nii võimalikud vähimad kui ka võimalikud suurimad.

Element on korraga võimalik vähim ja võimalik suurim ainult siis, kui teda veel millegagi võrreldud ei ole.

Seega väheneb võimalike vähimate ja suurimate elementide arv 2 võrra ülimalt $\lfloor n/2 \rfloor$ võrdlemisel. Teistel võrdlemistel väheneb ta ülimalt 1 võrra.

Vaja on seega vähemalt $\lfloor n/2 \rfloor + (2n - 2 - 2\lfloor n/2 \rfloor) = 2n - 2 - \lfloor n/2 \rfloor = n + \lceil n/2 \rceil - 2 = \lceil 3n/2 \rceil - 2$ võrdlemist.

Sarnase mõttekäiguga („pahatahtlik oraakel") saab leida näiteks ka järgmised alamtõkked:

Kui mitme võrdusekontrolliga „$a_i = a_j$?" on võimalik kindlaks teha, kas massiivis $a_1, \ldots, a_n$ leidub kaks võrdset elementi?

Antud suunamata graaf, lubatud on kontrollida, kas mingi kahe tipu vahel leidub serv. Kui mitu kontrolli on vajalik selleks, et teha kindlaks, kas graaf on sidus?

(Vastused: mõlemal juhul tuleb kõik paarid läbi kontrollida.)

Kuidas leida massiivi $a_1, \ldots, a_n$ k-ndat elementi mittekahanemise järjekorras?

Selleks võime leida k vähimat elementi.

```
for i := 1 to k do
    m_i := ∞
for i := 1 to n do
    if a_i < m_k then
        j := k − 1
        while j ⩾ 1 and a_i < m_j do
            m_{j+1} := m_j
            j := j − 1
        m_{j+1} := a_i
return m_k
```

Siin leidsime k-nda elemendi, mitte tema indeksi.

Keerukus:

- Kaks tsüklit teineteise sees.
  - Välimist täidetakse n korda.
  - Sisemist täidetakse kuni k korda.
- Seega keerukus halvimal juhul $\Theta(nk)$.

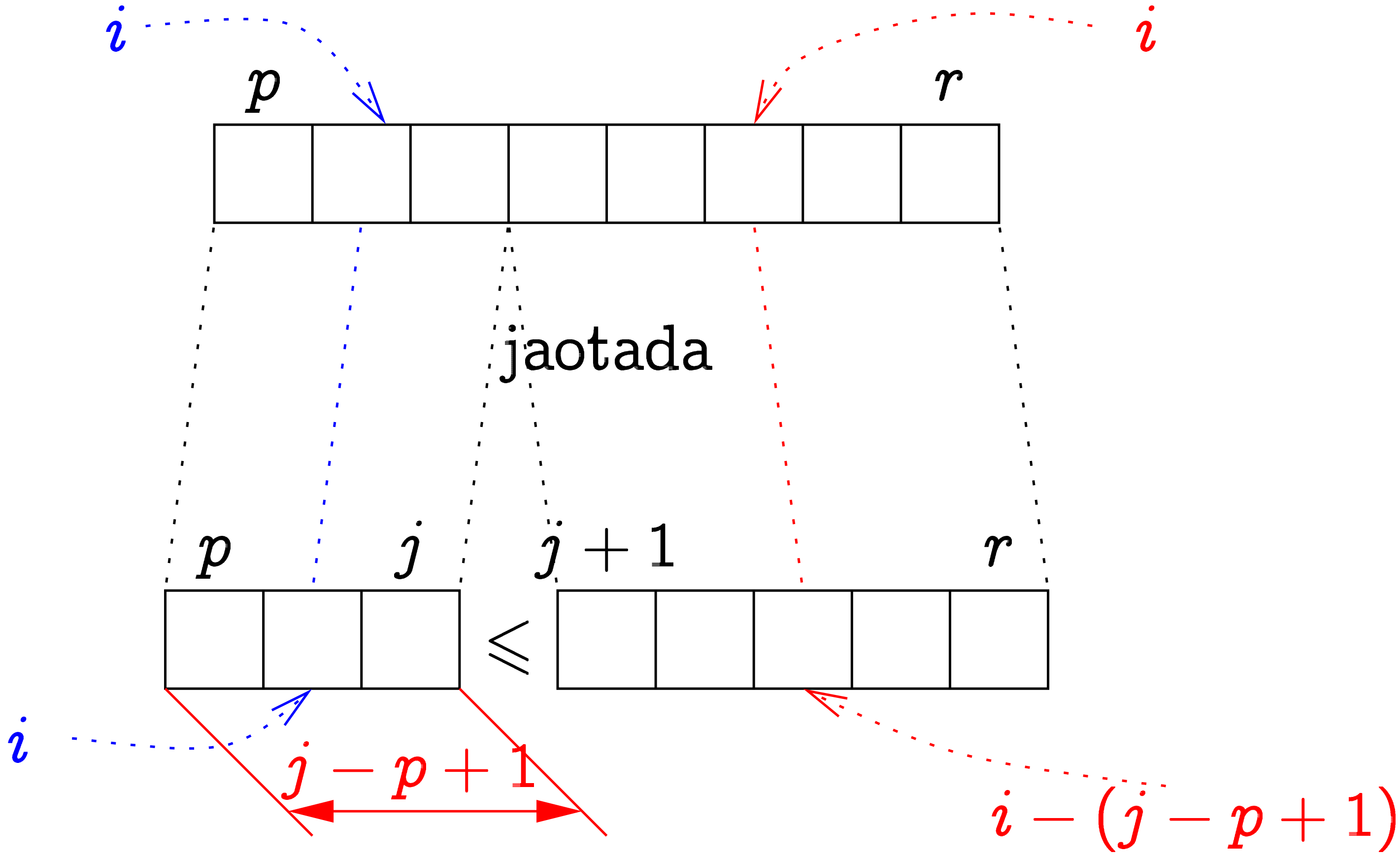
k-nda elemendi leidmist on võimalik teha ka keskmise keerukusega $\Theta(n)$. Meetod meenutab kiirsorteerimist.

randomiseeritud_vali(a, p, r, i) leiab massiivi a lõigust $p..r$ i-nda elemendi mittekahanemise järjekorras. Ta on:

if $p = r$ **then return** a_p
$j :=$ randomiseeritud_jaotada(a, p, r)
if $i \leqslant j - p + 1$ **then**
    return randomiseeritud_vali(a, p, j, i)
else
    return randomiseeritud_vali$(a, j + 1, r, i - j + p - 1)$

Eeldatud on, et $1 \leqslant i \leqslant r - p + 1$

k-nda elemendi valimiseks: randomiseeritud_vali$(a, 1, n, k)$.



i
i
p
r
jaotada
p
j
j + 1
r
⩽
i
j − p + 1
i − (j − p + 1)

Olgu $T(n)$ randomiseeritud_vali keskmine keerukus massiivil suurusega n. Siis

$$T(n) \leqslant \sum_{i=1}^{n-1} \lambda_i \max(T(i), T(n-i)) + \Theta(n),$$

kus λ_i on tõenäosus, et randomiseeritud_jaotada tagastab i.

Eelmises loengus leidsime, et $\lambda_1 = 2/n$ ja $\lambda_i = 1/n$, kui $i > 1$.

$$T(n) \leqslant \frac{1}{n}\left(\max(T(1), T(n-1)) + \sum_{i=1}^{n-1} \max(T(i), T(n-i))\right) + \Theta(n)$$

Ilmselt on T monotoonselt kasvav.

$$T(n) \leqslant \frac{1}{n}\left(T(n-1) + 2\sum_{i=\lceil n/2\rceil}^{n-1} T(i)\right) + \Theta(n)$$

$$= \frac{2}{n}\sum_{i=\lceil n/2\rceil}^{n-1} T(i) + \Theta(n)$$

Siin oleme liikme $\frac{1}{n}T(n-1)$ võtnud liikme $\Theta(n)$ sisse.

Näitame, et leidub $c \in \mathbb{N}$ nii, et $T(n) \leqslant cn$. Induktsioon üle n-i.

Kui $n = 1$, siis kui võtame $c \geqslant T(1)$, siis võrratus kehtib.

Olgu $n > 1$. Siis…

$$
\begin{aligned}
T(n) &\leqslant \frac{2}{n} \sum_{i=\lceil n/2 \rceil}^{n-1} T(i) + \Theta(n) \\
&\leqslant \frac{2}{n} \sum_{i=\lceil n/2 \rceil}^{n-1} ci + \Theta(n) \\
&= \frac{2c}{n} \cdot \frac{\lceil n/2 \rceil + n - 1}{2}((n-1) - \lceil n/2 \rceil + 1) + \Theta(n) \\
&\leqslant \frac{2c}{n} \cdot \frac{3n}{4} \cdot \frac{n}{2} + \Theta(n) \\
&= \frac{3}{4}cn + \Theta(n) \leqslant cn,
\end{aligned}
$$

sest c võib võtta piisavalt suure selleks, et $cn/4$ oleks suurem kui liige $\Theta(n)$.

Oleme näidanud, et $T(n) = O(n)$. Ilmselt ka $T(n) = \Omega(n)$, sest juba vähimagi elemendi leidmiseks läheb vaja vähemalt $n - 1$ võrdlemist.

Massiivi $a_1, \ldots, a_n$ suuruselt i-ndat elementi saab leida ka deterministlikult, halvima ajalise keerukusega $O(n)$.

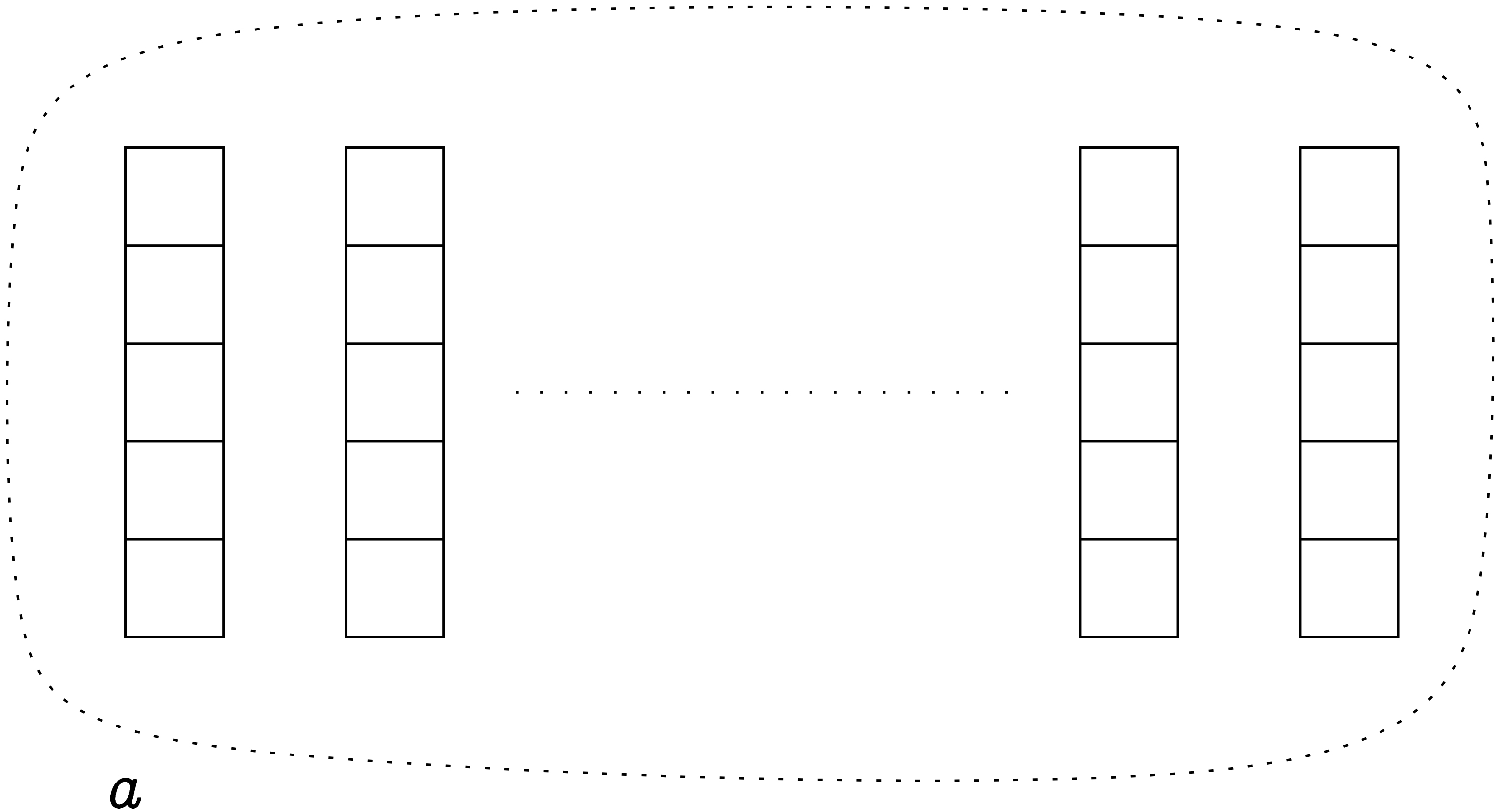
Olgu jaotada'(a, i, j, b) defineeritud nagu jaotada(a, i, j) joonisel 4.4, selle vahega, et suurust b ei leita mitte protseduuri sees, vaid ta on juba parameetrina ette antud.

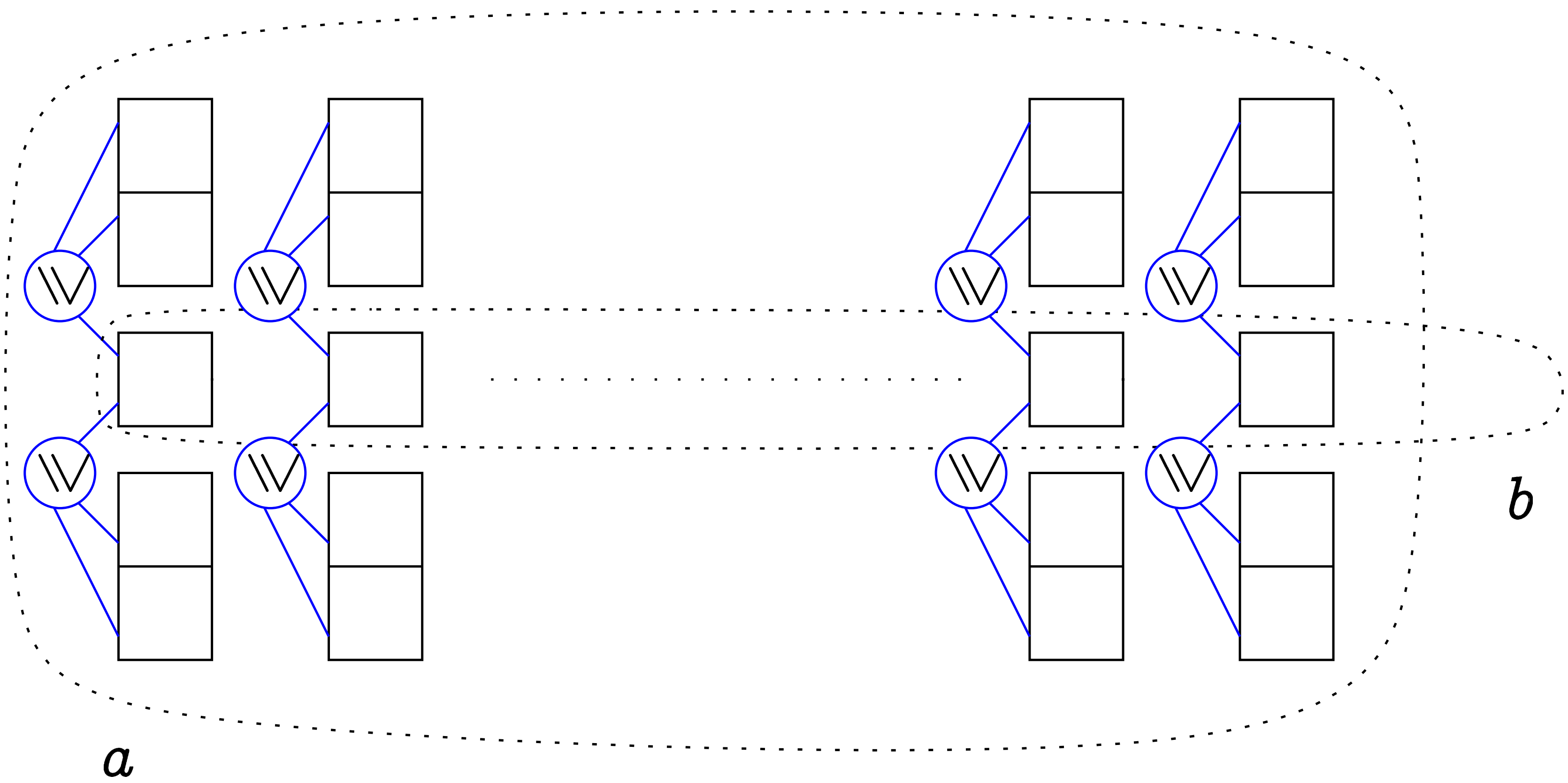
Funktsioon vali(a, p, r, i) leiab massiivi a lõigust $p..r$ i-nda elemendi mittekahanemise järjekorras.

Jagame $a_p, \ldots, a_r$ viieelemendilisteks gruppideks (viimane võib väiksem olla), leiame iga grupi keskmise elemendi (ja salvestame massiivi b, mis on funktsiooni „vali“ lokaalseks muutujaks).

```
if p = r then return a_p
pp := p; k := 0
loop
    gp := min(4, r − pp); k := k + 1
    sorteerida(a, pp, pp + gp)        --- meetod pole oluline
    b_k := a_{pp+⌈gp/2⌉}
    pp := pp + 5
    if pp > r then break
  ...
```

Sorteerimismeetod pole oluline, sest sorteerime konstantse pikkusega lõikusid, s.t. ühe sorteerimise keerukus on $\Theta(1)$.


a


a
b

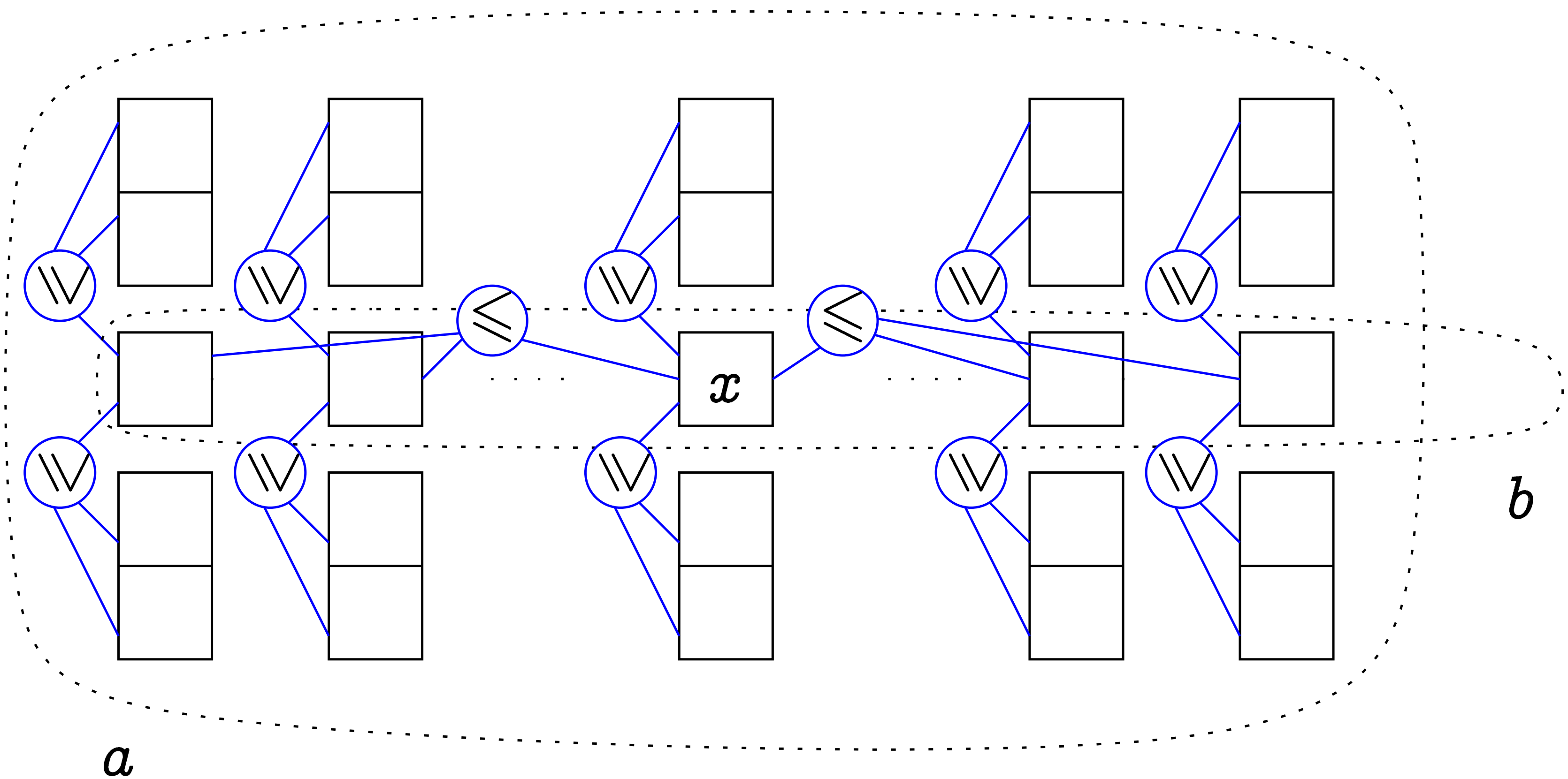
Leia keskmiste elementide keskmine element ja võta see veelahkmeks.

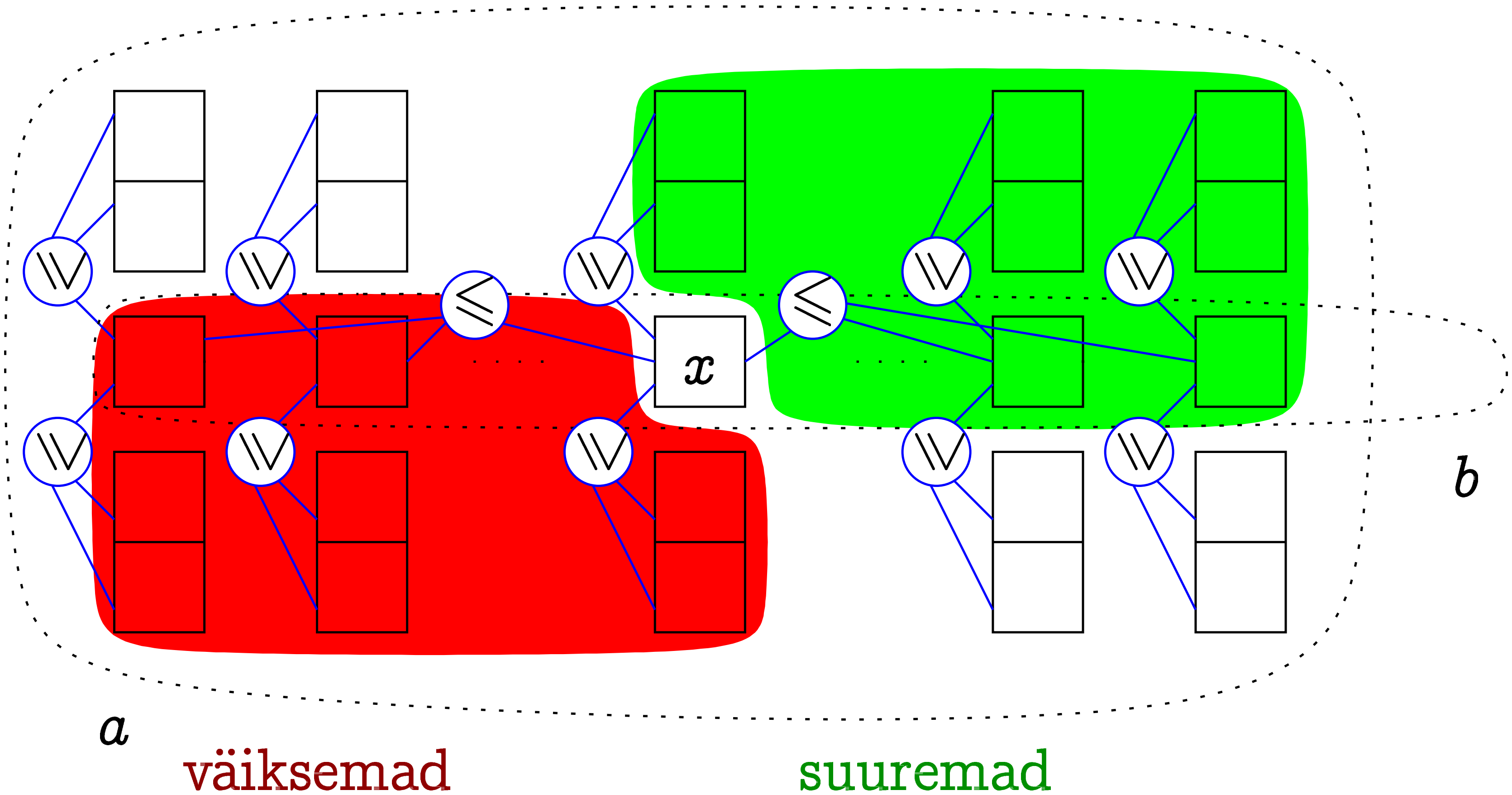
$x := \mathrm{vali}(b, 1, k, \lceil k/2 \rceil)$
$j := \mathrm{jaotada'}(a, p, r, x)$

Jätka, nagu funktsioonis randomiseeritud_vali.

if $i \leqslant j - p + 1$ **then**
    return $\mathrm{vali}(a, p, j, i)$
else
    return $\mathrm{vali}(a, j + 1, r, i - j + p - 1)$

Väide: x on hea veelahe, s.t. temast väiksemaid elemente on $\Theta(n)$ ja suuremaid samuti $\Theta(n)$.


x
a
b



x
a
b
väiksemad
suuremad

Näeme, et x-st väiksemaid elemente on vähemalt $\frac{3}{10}n - k$, kus k on mingi konstant.

Tõepoolest, viiestest gruppidest umbes pooltedel on vähemalt kolm elementi (kahel grupil võib olla vähem kui kolm) väiksemad kui x. Neid gruppe on $n/10$ tükki.

Liige „$-k$“ on tingitud sellest, et kahes vaadeldavas grupis (grupis, kuhu kuulub x ja grupis, kus on $\leqslant 5$ elementi) võib olla vähem kui kolm elementi x-st väiksemad.

Me võiksime k leida, aga analüüsi jaoks pole see oluline.

Samuti on x-st suuremaid elemente vähemalt $\frac{3}{10}n - k$.

„vali“ rekursiivsel väljakutsel (rida 12&14) on töödeldava lõigu pikkus ülimalt $\frac{7}{10}n + k$.

Seega $T(n) \leqslant T(\lceil n/5 \rceil) + T(\frac{7}{10}n + k) + \Theta(n)$.

Näitame induktsiooniga üle n, et leidub c nii, et $T(n) \leqslant cn$. Teeme kõigepealt induktsiooni sammu. Olgu n piisavalt suur.

$$T(n) \leqslant c\lceil n/5 \rceil + c(7n/10 + k) + \Theta(n) \leqslant$$
$$cn/5 + c + 7cn/10 + kc + \Theta(n) = 9cn/10 + (k+1)c + \Theta(n) =$$
$$cn - \big(c(n/10 - (k+1)) - \Theta(n)\big)$$

Kui $n > 10(k+1)$, siis on $c(n/10 - (k+1))$ positiivne ja me võime valida c piisavalt suure selleks, et sulgudes olev liige oleks $\geqslant 0$.

Peale selle valime c vähemalt sama suure kui $\max\limits_{1 \leqslant n \leqslant 10(k+1)} \frac{T(n)}{n}$ (induktsiooni baas). Siis $T(n) \leqslant cn$ iga n jaoks.

Sorteerimine kiirmeetodil ajalise keerukusega $\Theta(n \log n)$:

kiir_sorteeri(a, p, r) on:

```
if p = r then return
pp := p; k := 0
loop
    gp := min(4, r − pp); k := k + 1
    sorteerida(a, pp, pp + gp)
    b_k := a_{pp+⌈gp/2⌉}
    pp := pp + 5
    if pp > r then break
x := vali(b, 1, k, ⌈k/2⌉)
j := jaotada'(a, p, r, x)
kiir_sorteeri(a, p, j)
kiir_sorteeri(a, j + 1, r)
```

Tõepoolest, eelmises loengus nägime, et kui jaotamisel on mõlema poole pikkused proportsionaalsed kogu massiivi pikkusega, siis tööaeg on $\Theta(n \log n)$.

Antud juhul see nii on. Samuti on jaotamise keerukus (koos sellele eelneva veelahkme valikuga) lineaarse keerukusega.