

Programmeerimine

16. loeng

Täna loengus

- Algoritmid ja keerukus
- Lineaarne vs. kahendotsimine
- Eksamist

Algoritmid ja keerukus

- Konkreetse programmi efektiivsus sõltub:
 - programmis kasutatud algoritmidest;
 - sisendandmete suuruselt;
 - aga näiteks ka konkreetse masina riistvarast;
 - ...
- Põhiidee: ignoreerime masinast sõltuvaid konstante ja uurime **asümptootilist** keerukust:
 - millise funktsiooni $T(n)$ järgi kasvab algoritmis tehtavate sammude arv, kui sisendi suurus n kasvab lõpmatuks.
- Asümptootilist keerukust tähistame $\mathcal{O}(T(n))$.
- Keerukuse tähistamisel piisab kui näidata ainult kõige "kõrgemat järku" term.
 - Näiteks: $2n^3 + 5n^2 + 12n + 4 = \mathcal{O}(n^3)$.

Algoritmid ja keerukus

Ülesande lahendamise aeg

Kui palju aega kulutab kaasaegne arvuti aega ülesande lahendamiseks sõltuvalt algandmete hulgast?

	10	20	30	40	50	60
n	0,00001	0,00002	0,00003	0,00004	0,00005	0,00006
n^2	0,0001	0,0004	0,0009	0,0016	0,0025	0,0036
n^3	0,001	0,008	0,027	0,064	0,125	0,216
n^5	0,1	3,2	24,3	1,7m	5,2m	13m
2^n	0,001	1,0	17,9m	12,7p	35,3a	366saj
3^n	0,059	58m	6,5a	3855saj	$2 * 10^8$ saj	10^{13} saj

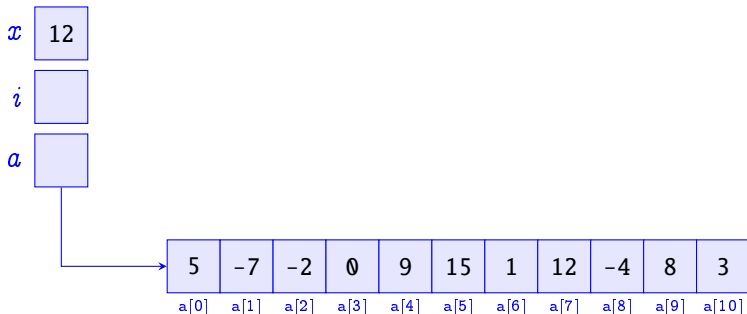
Massiivist elemendi otsimine

- Lihtsaim meetod massiivist elemendi otsimiseks vaatab massiivi elemente ükshaaval järjestikku läbi.
- Kui jõutakse otsitava elemendini, siis väljastatakse vastava elemendi indeks.
- Kui otsitavat elementi ei leidu, siis väljastatakse -1.

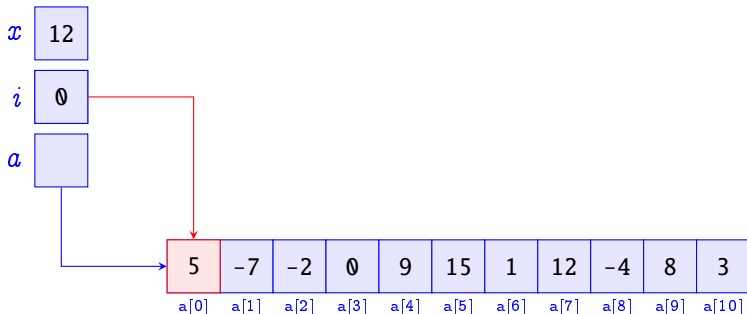
Lineaarne otsimine

```
def linearSearch(x, a):  
    for i in range(len(a)):  
        if a[i] == x: return i  
    return -1
```

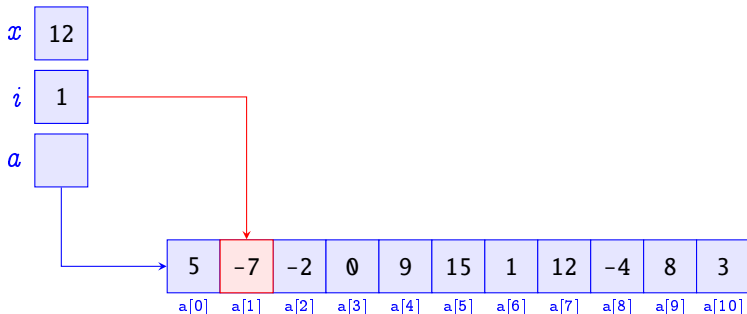
Massiivist elemendi otsimine



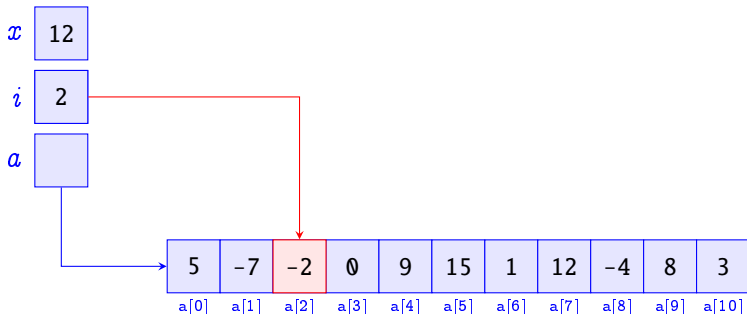
Massiivist elemendi otsimine



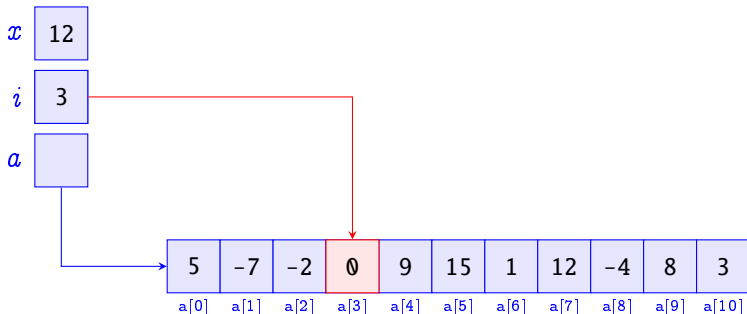
Massiivist elemendi otsimine



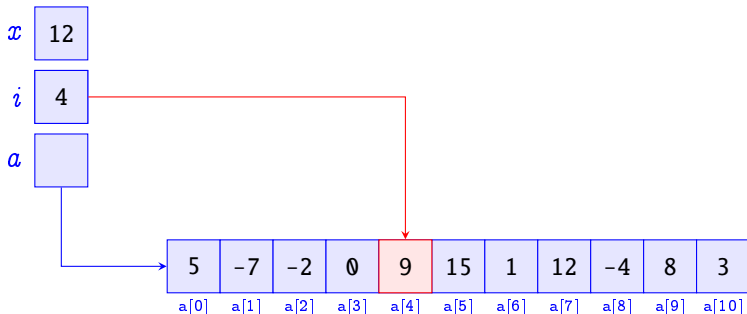
Massiivist elemendi otsimine



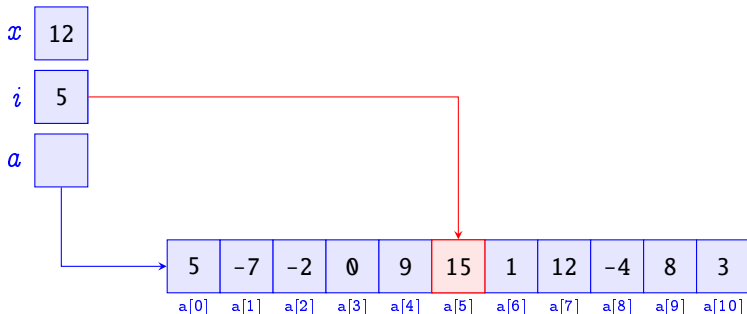
Massiivist elemendi otsimine



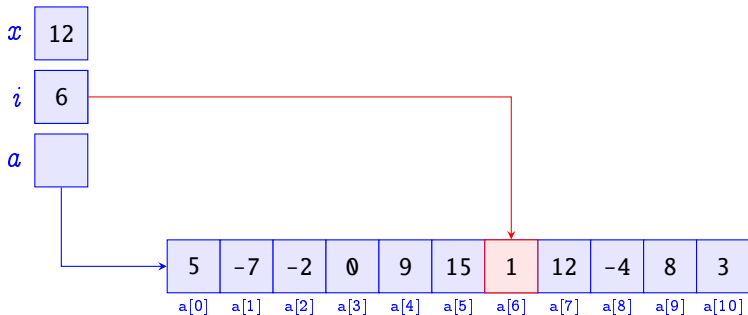
Massiivist elemendi otsimine



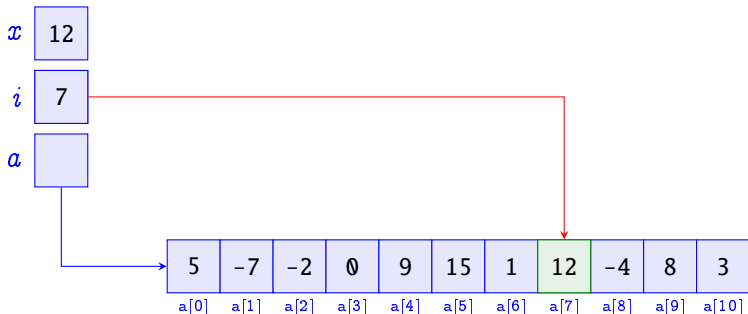
Massiivist elemendi otsimine



Massiivist elemendi otsimine



Massiivist elemendi otsimine



Massiivist elemendi otsimine

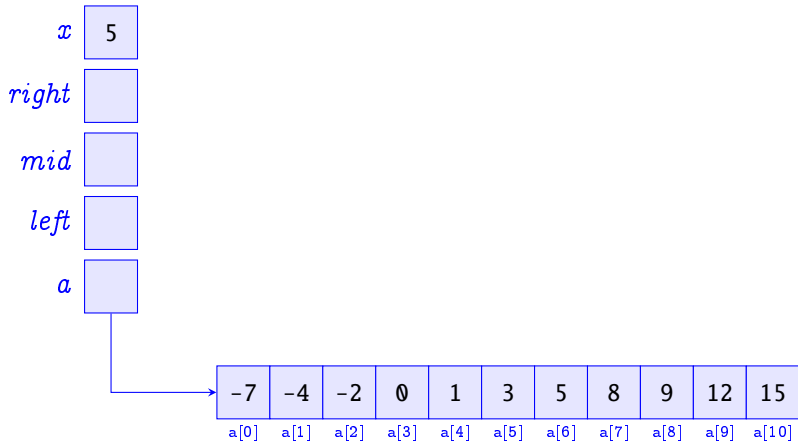
- Antud meetod on lineaarse keerukusega $\mathcal{O}(n)$, kuna halvimal juhul läbitakse kogu massiiv.
- Kui sisendmassiiv on *järjestatud*, siis on võimalik otsida tunduvalt efektiivsemalt kasutades **kahendotsimist**.
- Kahendotsimise idee:
 - Kontrollime massiivi keskmist elementi.
 - Kui see on otsitav, siis olemegi elemendi leidnud.
 - Kui see on otsitavast väiksem, siis otsitav saab olla ainult järgmiste elementide hulgas.
 - Vastasel korral saab otsitav olla ainult eelnevate elementide hulgas.
 - Kordame kogu protsessi vastaval alammassiivil.

Kahendotsimine

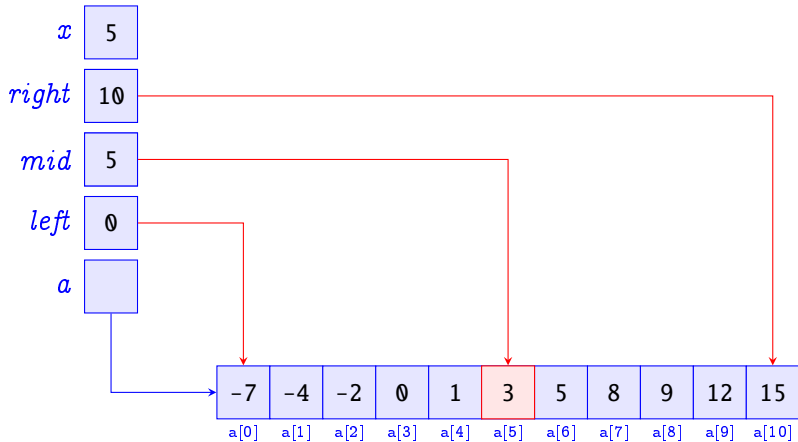
Kahendotsimine

```
def recBinSearch(x, a, left, right):  
    if left > right:  
        return -1  
    mid = (left+right) // 2  
    if a[mid] == x:  
        return mid  
    elif a[mid] > x:  
        return recBinSearch(x, a, left, mid-1)  
    else:  
        return recBinSearch(x, a, mid+1, right)  
  
def binarySearch(x, a):  
    return recBinSearch(x, a, 0, len(a)-1)
```

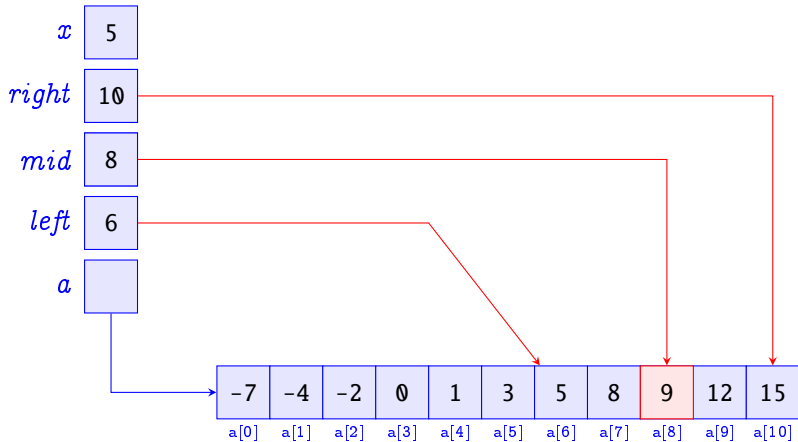
Kahendotsimine



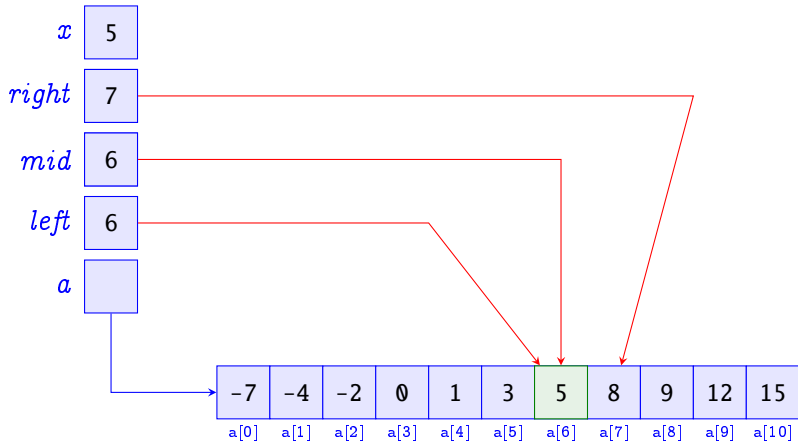
Kahendotsimine



Kahendotsimine



Kahendotsimine



Kahendotsimine

- Igal iteratsioonisammul välistatakse pool massiivist.
- Seega on kahendotsimine logaritmilise keerukusega $\mathcal{O}(\log_2(n))$.
- Suurte massiivide korral on erinevus efektiivsuses väga märgatav.
 - Näiteks, kui massiivis on $2^{20} = 1048576$ elementi, siis lineaarse otsimisega tuleb halvimal juhul kogu massiiv läbida.
 - Kahendotsimise korral tuleb halvimal juhul teha vaid 20 võrdlust.
- **NB!** Kahendotsimine eeldab, et sisendmassiiv on järjestatud; lineaarne otsimine seda ei eelda.

Eksamist

Eksamist

- 2. vaheeksami toimumisajad/kohad:
 - Teisipäev, 05. jaanuar 2016, kell 10.00-13.00
 - Esmaspäev, 11. jaanuar 2016, kell 10.00-13.00
 - Toimumiskoht Liivi 2, klassid 004, 202, 203, 205, 404.
- Lõputesti toimumisajad:
 - Reede, 08. jaanuar 2016, kell 10.00-12.00
 - Neljapäev, 14. jaanuar 2016, kell 10.00-12.00
 - Toimumiskoht Liivi 2, klassid 122/(207,403) 206, 402, 404, 405.
- Vaheeksamile ja lõputestile tuleb eelregistreerida õppeinfosüsteemi kaudu.
- Lõputest on kirjalik ...
 - ... ehk siis, kaasa tuleb võtta midagi, mis on sobilik paberi kirjutamiseks (näit. pastapliiats).
- Abimaterjalide kasutamine ei ole lubatud.

Hinde kujunemine

- Hinde kujuneb praktikumides ja eksamil kogutud punktide summana.
- Praktikumid (kuni 30 punkti):
 - koduülesanded (10 punkti)
 - tärnülesanded (10 punkti)
 - projekt (10 punkti)
- Eksam (kuni 80 punkti)
 - 2 vaheeksamit ($20 + 20 = 40$ punkti)
 - lõputest (40 punkti)
- NB! Igast eksamiosast tuleb saada vähemalt 50%.

Hinde skaala

	Hinne	Punkte
A	suurepärase	≥ 91
B	väga hea	81 – 90
C	hea	71 – 80
D	rahuldav	61 – 70
E	kasin	51 – 60
F	puudulik	0 – 50

Järeleksamist

- Järeleksamid toimuvad 20.-25. jaanuaril 2016:
 - 20. jaan. 2016 1. vaheeksam
 - 21. jaan. 2016 2. vaheeksam
 - 25. jaan. 2016 lõputest
- Ka järeleksamile on vaja registreeruda ...
 - ... aga alles/ainult siis, kui põhieksam ebaõnnestub.
- Vaja on sooritada ainult see osa, mis esialgselt ebaõnnestus.