

KĀRTOŠANA UN MEKLĒŠANA I 40.nodarbība

KUR MĒS SASTOPAMIES AR INFORMĀCIJAS MEKLĒŠANU?

- Bibliotēkā
- Vārdnīcā
- Tālrunī
- Google
- Meklēt palīdz katalogi un datorprogrammas



KĀPĒC INFORMĀCIJA JĀKĀRTO?

Ja dati ir sakārtoti, varam tajos ātrāk atrast nepieciešamo informāciju.

Kontakti tālrunī:

Anna 26262626

Apse Lauris 29292929

...

Kalns Ivo 61616161

Kalsons 11111111

MEKLĒŠANAS ALGORITMI

- Pilnā pārļase jeb lineārā meklēšana:
 - der visiem gadījumiem,
 - lēna, ja daudz datu.
- Intervāla dalīšana uz pusēm jeb binārā meklēšana:
 - der tikai sakārtotiem datiem,
 - ievērojami ātrāka.

1. SITUĀCIJA INFORMĀCIJAS MEKLĒŠANĀ

- Nesakārtoti dati bez atkārtojumiem.
- Meklējam, kam bija jāpērk *konservi*.
- Pēc 6.pārbaudes esam atraduši.
- Varēja būt arī 7 pārbaudes.
- Vispārīgā gadījumā n pārbaudes.

nr	vārds	pirkums
1	Ivo	maize
2	Dana	desa
3	Ilze	siers
4	Toms	makaroni
5	Olga	tēja
6	Kirils	konservi
7	Lara	gurķi

2. SITUĀCIJA INFORMĀCIJAS MEKLĒŠANĀ

- Nesakārtoti dati ar atkārtojumiem.
- Meklējam, kas bija jāpērk *Danai*.
- Pēc 2.pārbaudes esam atraduši.
- Taču jāmeklē tālāk līdz pat beigām.
- Vispārīgā gadījumā n pārbaudes.

nr	vārds	pirkums
1	Ivo	maize
2	Dana	desa
3	Ilze	siers
4	Ivo	makaroni
5	Olga	tēja
6	Kirils	konservi
7	Dana	gurķi

3. SITUĀCIJA INFORMĀCIJAS MEKLĒŠANĀ

- Sakārtoti dati bez atkārtojumiem.
- Meklējam, kam bija jāpērk *konservi*.
- Pēc 3.pārbaudes esam atraduši.
- Taču varēja gadīties, ka jāmeklē *tēja*.
- Vispārīgā gadījumā n pārbaudes.

nr	vārds	pirkums
1	Ivo	desa
2	Dana	gurķi
3	Ilze	konservi
4	Ivo	maize
5	Olga	makaroni
6	Kirils	siers
7	Lara	tēja

4. SITUĀCIJA INFORMĀCIJAS MEKLĒŠANĀ

- Sakārtoti dati ar atkārtojumiem.
- Meklējam, kas bija jāpērk *Ivo*.
- Pēc 4.pārbaudes esam atraduši.
- Taču jāmeklē tālāk un varēja gadīties, ka jāmeklē *Olgas* pirkumi.
- Vispārīgā gadījumā n pārbaudes.

nr	vārds	pirkums
1	Dana	maize
2	Dana	desa
3	Ilze	siers

4	Ivo	makaroni
5	Ivo	tēja
6	Kirils	konservi
7	Olga	gurķi

LINEĀRĀS MEKLĒŠANAS ALGORITMA ĀTRDARBĪBA

- Algoritmu ātrdarbības novērtēšanai izmanto pierakstu $O(n)$.
- $O(n)$ nozīmē, ka strādājot ar n datu vienībām, būs jāveic n atkārtotas darbības, ņemot vērā sliktāko gadījumu.
- Lineārās meklēšanas algoritma ātrdarbība ir $O(n)$.

BINĀRĀ MEKLĒŠANA JEB INTERVĀLA DALĪŠANA UZ PUSĒM

0	1	2	3	4	5	6	7	8	9	10
Ann a	Dan a	Ivo	Jan a	Kim s	Koļ a	Olg a	Petr o	Pol s	Sār a	Tom s
2.3 6	4.5 0	0.8 0	5.1 2	2.8 0	4.0 0	1.1 0	0.98	4.0 5	0	2.5 8

- Sakārtoti dati bez atkārtojumiem.
- Meklējam, cik samaksāja Pols.
- Nesākam no sākuma, bet no vidējā
 $\text{vidējais} = (\text{kreisais} + \text{labais}) / 2$
- $\text{Pols} > \text{Koļa}$, tāpēc meklējamo apgabalu sašaurinām no 6. līdz 10. un tajā atrodam vidējo.
- Ar 2 salīdzināšanām esam atraduši meklēto.

JAUTĀJUMS

- Kā noteikt cik salīdzināšanas darbības sliktākajā gadījumā jāveic binārās meklēšanas laikā, ja doti 1000 uzvārdi?
- Cik reizes varam meklējumu apgabalu dalīt uz pusēm?

500->250->125->63->32->16->

8->4->2->1

- 10 reizes

BINĀRĀS MEKLĒŠANAS ALGORITMA NOVĒRTĒJUMS

- Binārās meklēšanas algoritma ātrdarbība ir $O(\log_2 n)$
- $2^{10}=1024$
- $\log_2 1024=10$
- $\log_2 1000 \approx 10$
- $\theta(n) < O(\log_2 n)$

KĀ NOVĒRTĒT CITU ALGORITMU ĀTRDARBĪBU?

- Darbības, kas atkārtojas vienā ciklā.
- Ātrdarbība=cikla atkārtojumu skaits.

```
1 import java.util.Random;
2 class Mazakais {
3     public static void main(String args[]) {
4         int n=100;
5         int skaits[] =new int[n];
6         Random rand=new Random();
7         for(int i=0; i<n; i++){
8             skaits[i] = rand.nextInt(200);
9             System.out.format("%4d",skaits[i]);
10        }
11        System.out.println();
12        int min = skaits[0];
13        for(int i=1;i<n;i++){
14            if (min>skaits[i])min=skaits[i];
15        }
16        System.out.println("Mazakais skaitlis ir "+min);
17    }
18 }
```

RESURSI INTERNETĀ

Izmantotā saite:

<http://www.math.cornell.edu/~dtaimina/Artexhibits.htm>

Ko vēl var palasīt:

<http://www.google.com/intl/lv/insidesearch/howsearchworks/algorithms.html>

<http://www.programcreek.com/2014/04/check-if-array-contains-a-value-java/>