

Algoritmusok és adatszerkezetek

Göncziné Kapros Katalin
gonczone.kapros.katalin@gde.hu

Gyak



Adatsorhoz egy adatot rendelő eljárások Keresések

Teljes keresés

```
eljárás teljesKeresés
```

```
  I = 0
```

```
  ciklus, amíg (I < N) és (A[I] ≠ X)
```

```
    I = I + 1
```

```
  ciklus vége
```

```
  ha I < N, akkor
```

```
    ki: I
```

```
  különben
```

```
    ki: "A keresett elem nem található."
```

```
  elágazás vége
```

```
eljárás vége
```

Strázsás keresés

eljárás strázsásKeresés

$A[N] = X$

$I = 0$

ciklus, amíg $A[I] \neq X$

$I = I + 1$

ciklus vége

ha $I < N$, akkor

ki: I

különben

ki: "A keresett elem nem található."

eliárás vége

Lineáris keresés (sequential search)

- A keresés előfeltétele, hogy az adatsor rendezett legyen $a_0 \leq a_1 \leq a_2 \leq \dots \leq a_n$.
- Az eljárás lényege, hogy teljes keresést végzünk az első elemtől kezdve, de csak addig, amíg a keresett elem egyáltalán előfordulhat a tömbben.

Keresett: 25

5	6	9	9	11	17	23	26	29	32	34
0	1	2	3	4	5	6	7	8	9	10

Lineáris keresés (sequential search)

- Mivel az adatsor rendezett, megállhatunk, ha az $A[I] < X$ feltétel már nem teljesül. Vagy azért mert $A[I] = X$, vagy mert a tömb I -nél nagyobb indexű elemei mind nagyobbak, mint X , tehát nincs közte.

```
eljárás lineárisKeresés
```

```
    I = 0
```

```
    ciklus, amíg (I < N) és (A[I] < X)
```

```
        I = I + 1
```

```
    ciklus vége
```

```
    ha (I < N) és (A[I] = X), akkor
```

```
        ki: I
```

```
    különben
```

```
        ki: "A keresett elem nem található."
```

```
eljárás vége
```

Bináris (logaritmikus) keresés (binary search)

- A bináris keresés alkalmazásának szintén előfeltétele, hogy az adatsor rendezett legyen.

A	7	8	10	10	17	21	23	23	31	37	40	49	61	67	73
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	7	8	10	10	17	21	23	23	31	37	40	49	61	67	73
1.	↑ E=0							↑ K=7							↑ U=14
2.									↑ E=8			↑ K=11			↑ U=14
3.									↑ E=8	↑ K=9	↑ U=10				

- E a tartomány első eleme,
- K a tartomány középső eleme
- U a tartomány utolsó eleme,
- X keresett elem $X=37$

Bináris keresés, ha nincs a keresett elem

- Legyen $X = 9$

<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>	<i>13</i>	<i>14</i>
7	8	10	10	17	21	23	23	31	37	40	49	61	67	73

$$1. \quad \begin{array}{c} \uparrow \\ E=0 \end{array} \qquad \qquad \qquad \begin{array}{c} \uparrow \\ K=7 \end{array} \qquad \qquad \qquad \begin{array}{c} \uparrow \\ U=14 \end{array}$$

2. $\begin{matrix} \uparrow \\ E=0 \end{matrix}$ $\begin{matrix} \uparrow \\ K=3 \end{matrix}$ $\begin{matrix} \uparrow \\ U=6 \end{matrix}$

3. $E=0$ $K=1$ $U=2$

4. $E=K=U=0$

5. $\begin{matrix} \uparrow & \uparrow \\ U=1 & E=2 \end{matrix}$

Bináris keresés (binary search)

- Kimondhatjuk, hogy a bináris keresés legfeljebb $M = \lceil \log_2 N \rceil + 1$ lépésben megtalálja a keresett elemet, vagy megáll, ha a keresett elem nem található az adatsor-ban. Az $\lceil x \rceil$ jelölés x felső egészrészét jelenti.

Ha $N = 10$, akkor $M = \lceil \log_2 10 \rceil + 1 = 5$.

Ha $N = 100$, akkor $M = \lceil \log_2 100 \rceil + 1 = 8$.

Ha $N = 1000$, akkor $M = \lceil \log_2 1000 \rceil + 1 = 11$.

Ha $N = 10\,000$, akkor $M = \lceil \log_2 10\,000 \rceil + 1 = 15$.

Ha $N = 100\,000$, akkor $M = \lceil \log_2 100\,000 \rceil + 1 = 18$.

...

- Látható, hogy ha N értékét nagyságrendekkel növeljük is, a maximális lépések száma a keresésben csak minimálisan növekszik. Ezért ennek az eljárásnak igazán a nagy elemszámú adatsorokban nagy a hatékonysága.
- Természetesen nem szabad elfelednünk azt az előfeltételt, hogy az algoritmus rendezett adatsort igényel!

Bináris keresés (logaritmikus, binary search)

```
eljárás logaritmikusKeresés
```

```
  E = 0
```

```
  U = N - 1
```

```
  K = (E + U) / 2
```

```
  ciklus, amíg (E ≤ U) és (A[K] ≠ X)
```

```
    ha A[K] < X, akkor
```

```
      E = K + 1
```

```
    különben
```

```
      U = K - 1
```

```
    elágazás vége
```

```
    K = (E + U) / 2
```

```
  ciklus vége
```

```
  ha (E ≤ U), akkor
```

```
    ki: K
```

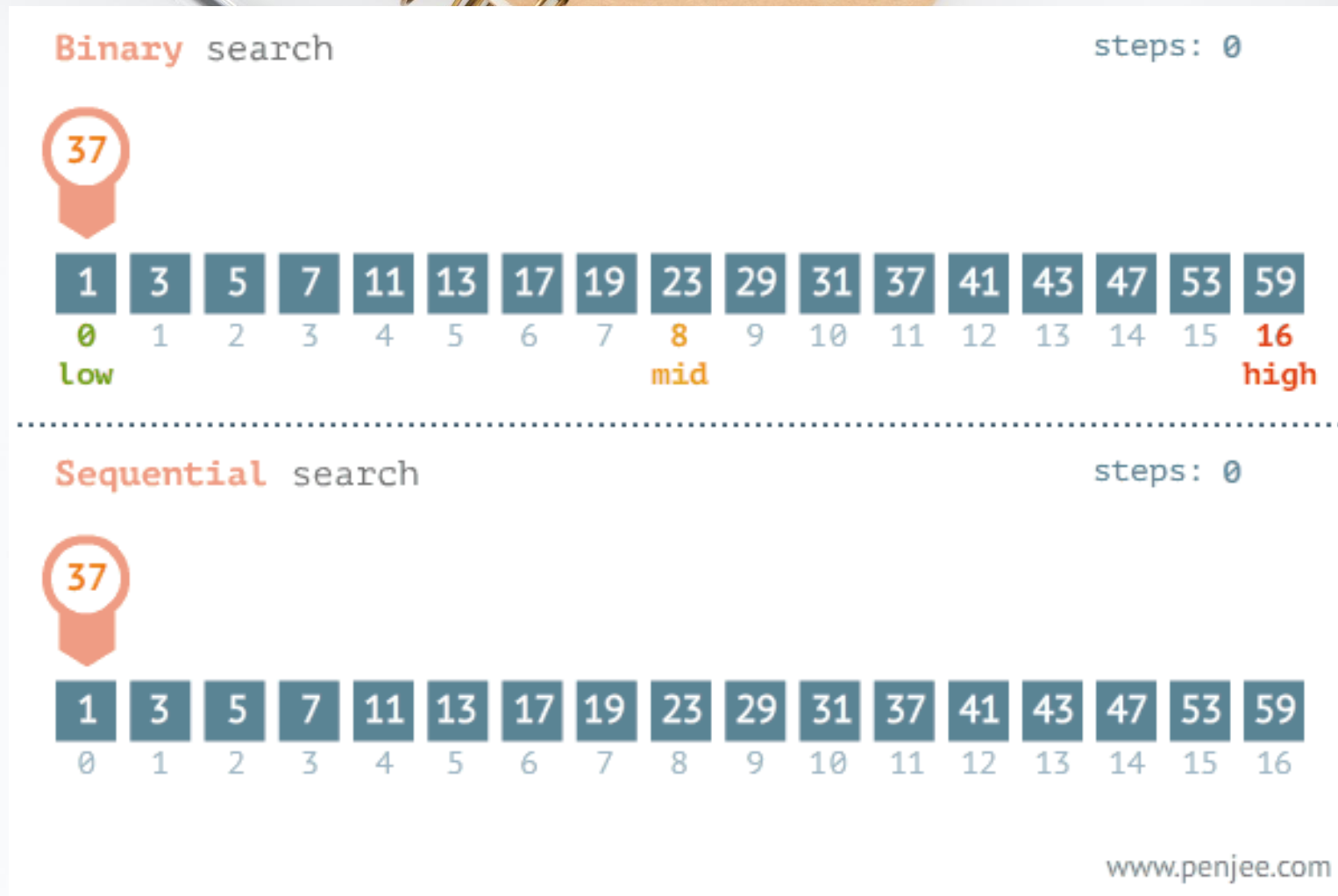
```
  különben
```

```
    ki: "A keresett elem nem található."
```

```
  elágazás vége
```

```
eljárás vége
```

Lineáris keresés vs Bináris keresés



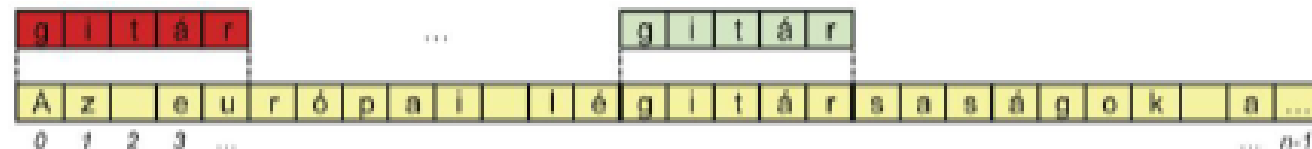
Sztring keresése sztringben – mintaillesztés

- **Mintaillesztés** (pattern matching): egy karakterlánc megkeresése egy másik karakterláncban
- Az a karakterlánc, amelyben keresünk, lehet akár egy teljes dokumentum is egy szövegszerkesztő programban. A feladat nemcsak sztringek esetén értelmezhető és használható, ugyanezt a problémát kell megoldani akkor is, ha egy szakaszt keresünk egy DNS-láncban.
- A keresett minta lehet reguláris kifejezés is. Ilyenkor nem egy konkrét sztinget keresünk, hanem egy a lehetséges sztringek alkotta halmazt (pl.: *.doc, szepes?.txt).
- Egyes – nem imperatív – programozási nyelvekben a mintaillesztés rendszerszinten támogatott szolgáltatás (pl. Perl).
- **„Brute force” eljárás**
- Hatékonysági mutatóit tekintve ez a leggyengébb módszer, viszont egyszerű és könnyen megérthető.
- Összetettebb mintaillesztő eljárások: KMP (Knuth–Morris–Pratt) algoritmus, Rabin–Karp algoritmus, Boyer–Moore algoritmus, Dömölki algoritmus.
- A brute force (nyers erő) vagy egyszerű mintaillesztési eljárás egy lineáris keresésbe ágyazott lineáris keresés. A sztringeket tekintjük olyan tömböknek, amelyek karaktereket tartalmaznak. Legyen A az a tömb, amelyben keresünk, B pedig a keresett sztring karaktereit tartalmazó tömb. Legyen N az A hossza, M pedig B hossza (karaktereinek száma). Így nyilvánvaló, hogy a legutolsó karakterpozíció, ahol az illeszkedés egyáltalán bekövetkezhet az $(N - M)$ pozíció.

Az alábbi algoritmus kiírja az első olyan kezdőpozíciót, ahol a B illeszkedik A-ra. Ha nincs ilyen, akkor pedig azt, hogy a keresett sztring nem található.

```
eljárás egyszerűMintaillesztés
  I = 0
  J = 0
  K = 0
  ciklus, amíg I ≤ (N - M) és J < (M - 1)
    ha A[K] = B[J], akkor
      K = K + 1
      J = J + 1
    különben
      I = I + 1
      J = 0
      K = I
    elágazás vége
  ciklus vége
  ha J = (M - 1), akkor
    ki: "Illeszkedik ezen a pozíción:", I
  különben
    ki: "Nincs illeszkedés."
  eljárás vége
```

Minta: gitár





Köszönöm a figyelmet!

Felhasznált szakirodalom

- Tömösközi Péter: Algoritmizálás alapjai EKF, 2011. ISBN 978-615-5221-14-9 (<https://docplayer.hu/10208397-Algoritmizalas-alapjai-tomoskozi-peter.html>)
- Dr. Geda Gábor: Adatszerkezetek és algoritmusok, EKF TTK, 2013 (<https://docplayer.hu/17242165-Adatszerkezetek-es-algoritmusok-geda-gabor.html>)
- Dr. Juhász István: Adatszerkezetek és algoritmusok jegyzet (<https://docplayer.hu/3509405-Adatszerkezetek-es-algoritmusok.html>)