

Eldöntés

- Az algoritmus eldönti, hogy van-e a tömbben adott tulajdonságú elem. Amint talál egyet, a ciklus leáll.
- Ha a ciklus azért állt le, mert túlléptünk a tömb utolsó, vizsgált elemén is, akkor nem volt benne keresett elem

6

```
• Van-e 50 az elemek között
i:=1
Ciklus amíg i<=N és T[i]<>50
i:=i+1
Ciklus vége
Ha i<=N akkor
ki: "volt 50"
Fontos feltételek sorrendje!!
```

Összegzés

- Egy tömb elemeinek összegzése
- Könnyen átírható szorzatra vagy más műveletre

```
s:=0
Ciklus i:=1...N
s:=s+T[i]
Ciklus vége
Ki: s
4
Megszámol
```

Kiválasztás

- Az algoritmus megadja, hogy a tömbben egy bizonyos elem hol (hányadik helyen) van.
- Csak akkor működik, ha biztosan van ilyen elem

```
i:=1
Ciklus amíg T[i]<>50
i:=i+1
Ciklus vége
ki: i
8
```

Keresés

Megszámlálás

- Megszámolja, hogy a tömbben hány, adott tulajdonságú elem van
- Például, negatív számok

```
s:=0
Ciklus i:=1..N
Ha T[i]<0 akkor s:=s+1
Ciklus vége
Ki: s
```

Lineáris keresés

- Az előzőnél biztonságosabb algoritmus: megadja, hogy van-e olyan elem, és ha igen, hányadik. (többféle kereső algoritmus van)

```
i:=1
Ciklus amíg i<=N és T[i]<>50
i:=i+1
Ciklus vége
Ha i<=N akkor
```

```
ki: i
különben
ki: -1 /* bármilyen más érvénytelen index */
```

Maximum kiválasztás

```
•
T tömb maximális elemének megkeresése
m:=1
Ciklus i:=2..N
Ha T[i]>T[m] akkor m:=i
Ciklus vége
Ki: m, T[m]
•
m: a pillanatnyilag talált legnagyobb elem helyét
mutatja
17
R
```