



# Adatbázis és szoftverfejlesztés

Algoritmus tervezés, végrehajtás szimulálása

# A feladat leírása

- Egy szervert üzemeltetünk
- Sok száz felhasználó léphet be a szerverre
- Ki kell szűrniük a logfájlból a rosszindulatúnak feltételezett ip címeket
- A logfájlban ilyen bejegyzések vannak:
  - 2014.09.03. 19:40 [10.0.0.1] Bejelentkezési kísérlet (user1)
  - 2014.09.03. 19:41 [10.0.0.1] Sikeres Bejelentkezés (user1)
  - 2014.09.03. 19:40 [10.0.0.2] Bejelentkezési kísérlet (user2)
  - 2014.09.03. 19:41 [10.0.0.2] Sikertelen Bejelentkezés (user2)
- Gyanúsak minősül az, ha kétszer egymás után sikertelenül jelentkezett be. Időben 1 telt el a két bejelentkezés között.

Kimenet: gyanús ip címek egy tömbben

# Mik kellenek?

- Amíg (feltétel)
- Ha (feltétel) akkor (utasítás) egyébként (utasítás)
- Sor = SorBeolvasás()
  - Sor[ip] = 1, 2, 3 ...
  - Sor[ido] = 1, 2, 3 ...
  - Sor[siker] = igaz/hamis
- NincsVégeAFájlnak – igaz, ha még van beolvasandó sor
- Eredmény[] – tömb, eredmények gyűjtése
- Sikertelen[] – tömb, ip-címekhez tartozó sikertelen kísérletek száma
- Lido[] – tömb, legutóbbi sikertelen bejelentkezési kísérlet ideje az adott ip-címről

# Első algoritmus

- Amíg ( NincsVégeAFájlnak )
  - Sor = sorBeolvasás()
  - ----- pillanatfelvétel, táblázatkitöltés -----
  - Ha sor[siker]
    - -
  - Egyébként
    - Ha sikertelen[sor[ip]]  $\geq 2$ 
      - Eredmény[] = sor[ip]
    - Egyébként
      - Sikertelen[sor[ip]]++

# Első algoritmus szimulációja

- 1 X
- 1 X
- 2 X
- 1 X
- 3 X
- 2 X

Eredmény[]	Holjárnk?	Sikertelen[1]	Sikertelen[2]	Sikertelen[3]
-	1. Sor	0	0	0
-	2. Sor	1	0	0
-	3.	2	0	0
-	4.	2	1	0
1	5.	2	1	0
1	6.	2	1	1
1	Vége	2	2	1

# Mi a hiba?

- Az algoritmus nem jó, mert a sikertelen számláló növelés előtt van az ellenőrzés és nem utána
- Így csak 2 körrel később került be az 1-es ip, és egyáltalán nem került be a 2-es.

# Második algoritmus

- Amíg ( NincsVégeAFájlnak )
  - Sor = sorBeolvasás()
  - ----- pillanatfelvétel, táblázatkitöltés -----
  - Ha sor[siker]
    - -
  - Egyébként
    - Sikertelen[sor[ip]]++
    - Ha sikertelen[sor[ip]] >= 2
      - Eredmény[] = sor[ip]

# Második algoritmus szimulációja

- 1 X
- 1 X
- 2 X
- 1 X
- 3 X
- 2 X

Eredmény[]	Holjárnk?	Sikertelen[1]	Sikertelen[2]	Sikertelen[3]
-	1. Sor	0	0	0
-	2. Sor	1	0	0
1	3.	2	0	0
1	4.	2	1	0
1,1	5.	3	1	0
1,1	6.	2	1	1
1,1,2	Vége	2	2	1



# Mi a hiba?

- Az algoritmus nem jó, mert többször is hozzáadódik az 1-es ip, pedig nem volt többször „gyanús” próbálkozása
- Nincs figyelembe véve az idő, ezért bekerül a 2-es ip, pedig nem kéne

# Harmadik algoritmus

- Amíg ( NincsVégeAFájlnak )
  - Sor = sorBeolvasás()
  - ----- pillanatfelvétel, táblázatkitöltés -----
  - Ha sor[siker]
    - -
  - Egyébként
    - Ha  $\text{sor}[\text{ido}] - \text{Lido}[\text{sor}[\text{ip}]] \leq 1$ 
      - Sikertelen[sor[ip]]++
      - Ha sikertelen[sor[ip]]  $\geq 2$ 
        - Eredmény[] = sor[ip]
    - Egyébként
      - Sikertelen[sor[ip]] = 0
    - Lido[sor[ip]] = sor[ido]

# Harmadik algoritmus szimulációja

- 1 x 1
- 1 x 2
- 2 x 3
- 1 x 4
- 3 x 5
- 2 x 6

Eredmény[]	Holjárnak?	Sikertelen[1]	Lido[1]	Sikertelen[2]	Lido[2]	Sikertelen[3]	Lido[3]
-	1. Sor	0	0	0	0	0	0
-	2. Sor	1	1	0	0	0	0
1	3.	0	1	0	0	0	0
1	4.	0	1	1	3	0	0
1	5.	1	4	1	3	0	0
1	6.	1	4	1	3	1	5
1	Vége	1	4	0	6	1	5