Rabin-Karp illesztő algoritmus

Marosvölgyi Gergely ©2014-2015

Bevezetés

A mintaillesztés egyik speciális esete, amikor az előforduló karakterek kizárólag decimális számjegyek lehetnek, vagyis $\sum = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Például:

Teljes sorozat: **6583139528 4079**

Keresett minta: **39528**

Ekkor célravezetőbb lehet, ha ezeket nem karakterláncként, hanem számként ábrázoljuk, és kihasználjuk az ezzel járó előnyöket.

Probléma

Az egyetlen probléma a számábrázolással, hogy megvannak a korlátai, ugyanis pl. egy előjel nélküli 32 bites egész szám legnagyobb értéke $2^{32}-1 = 4\ 294\ 967\ 295$ lehet.

64 bit esetén ez $2^{64}-1 = 18\ 446\ 744\ 073\ 709\ 551\ 615$.

Utóbbi már 20 jegyű szám, de mintaként tekintve rá még mindig rövid (20 karakternél hosszabbat is szeretnénk keresni).

A Rabin-Karp illesztő algoritmus éppen erre a problémára ad megoldást.

Ötlet

A teljes minta ábrázolása helyett vegyük inkább a minta osztási maradékát egy számmal, majd menjünk végig a teljes számsorozaton, és nézzük meg minden egyes mintányi hosszra, hol egyeznek meg a maradékok!

A nem egyező maradék egyben nem egyező mintát is jelent. Egyező maradék azonban nem feltétlenül jelent egyező mintát, hiszen több szám is adhatja ugyanazt a maradékot.

Kérdés, hogy milyen számmal (modulussal) célszerű venni az osztási maradékot...

Az ötlet szemléltetése

Teljes sorozat: **48391 5786**

Keresett minta: **391**

Az ötlet lényegének szemléltetéséhez vegyünk különböző modulusokkal osztási maradékot, pl. a 2, 7 és 19 számokkal.

```
391 mod 2  = 1

391 mod 7  = 6

391 mod 19 = 11
```

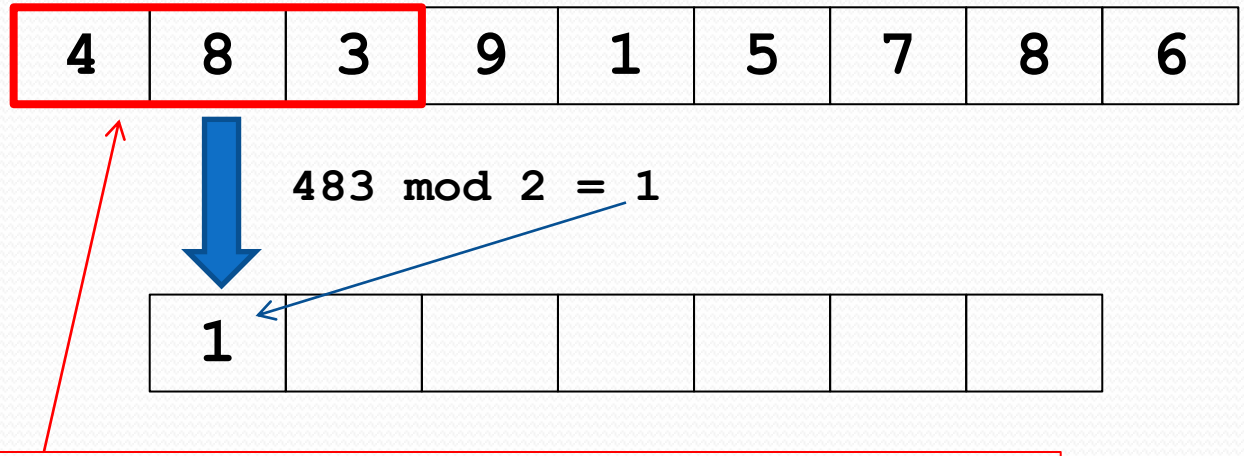
Tehát ha a 2-t választanánk, akkor az 1-es maradékot kellene figyelni, ha a 7-eset, akkor a 6-os maradékot, 19-nél pedig a 11-et. A továbbiakban egymás után vizsgáljuk ezeket, majd összevetjük az eredményeket.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:



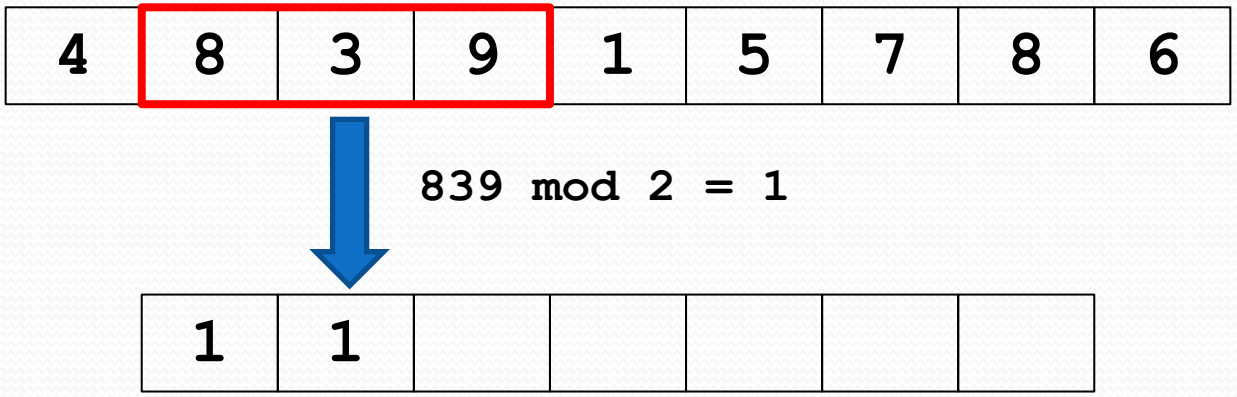
A minta hossza 3 (hiszen a ”391” 3-jegyű), ezért ugyanekkora méretű ablak által meghatározott számoknak fogjuk számolni a modulussal vett osztási maradékát.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

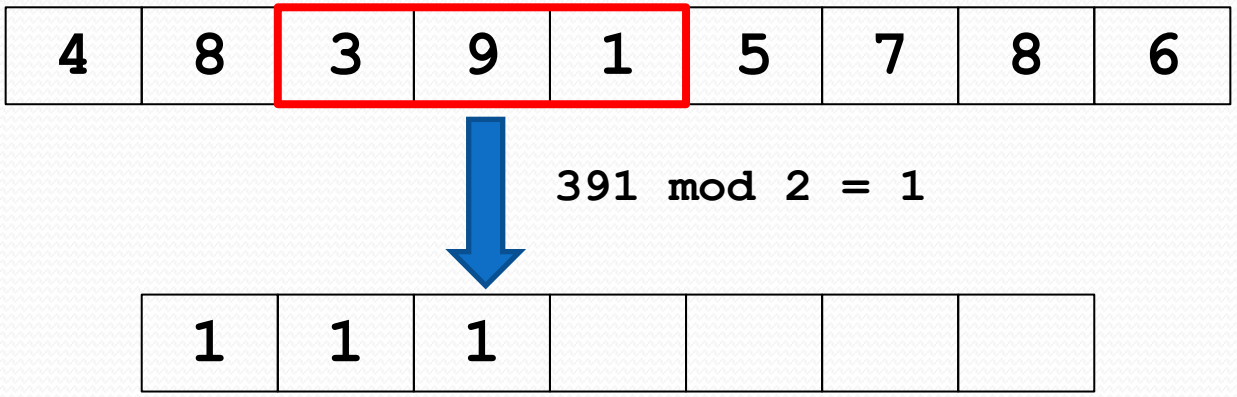


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

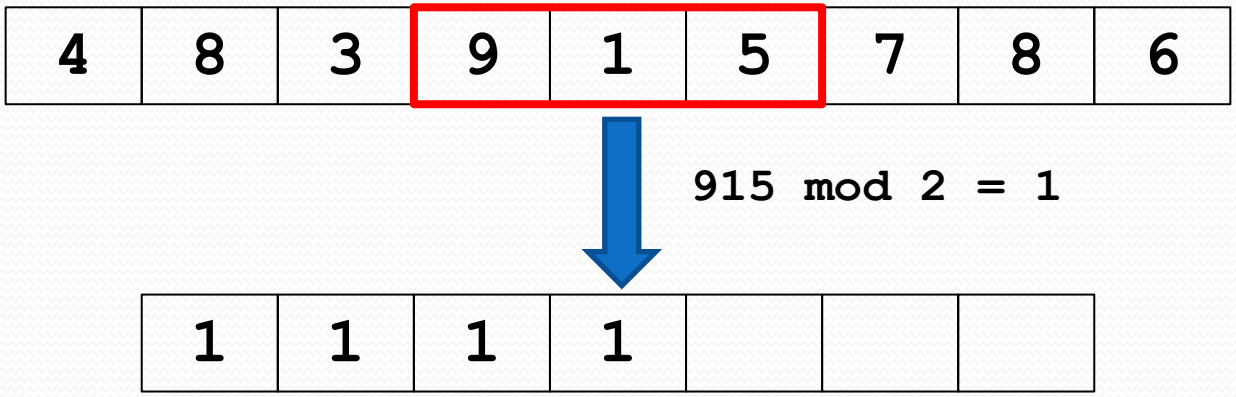


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839**15786

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

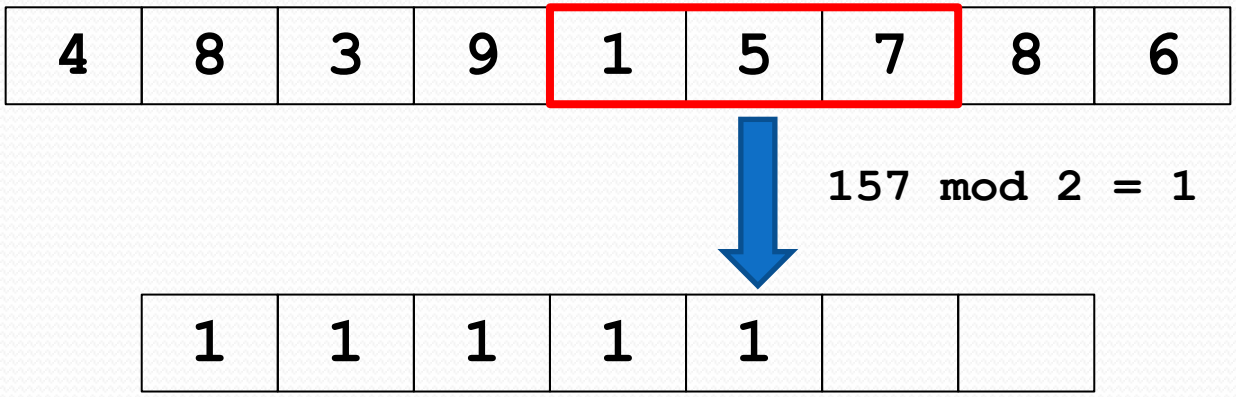


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

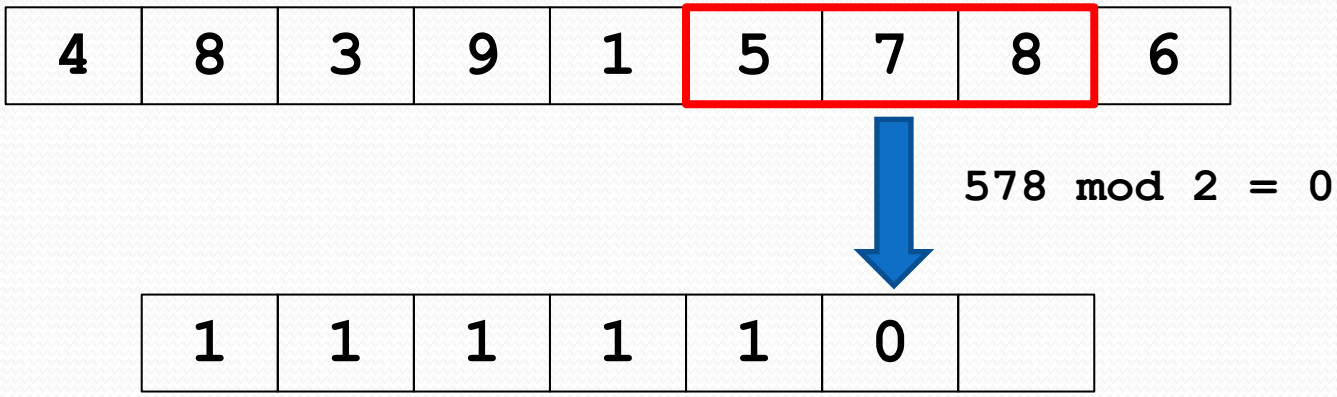


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

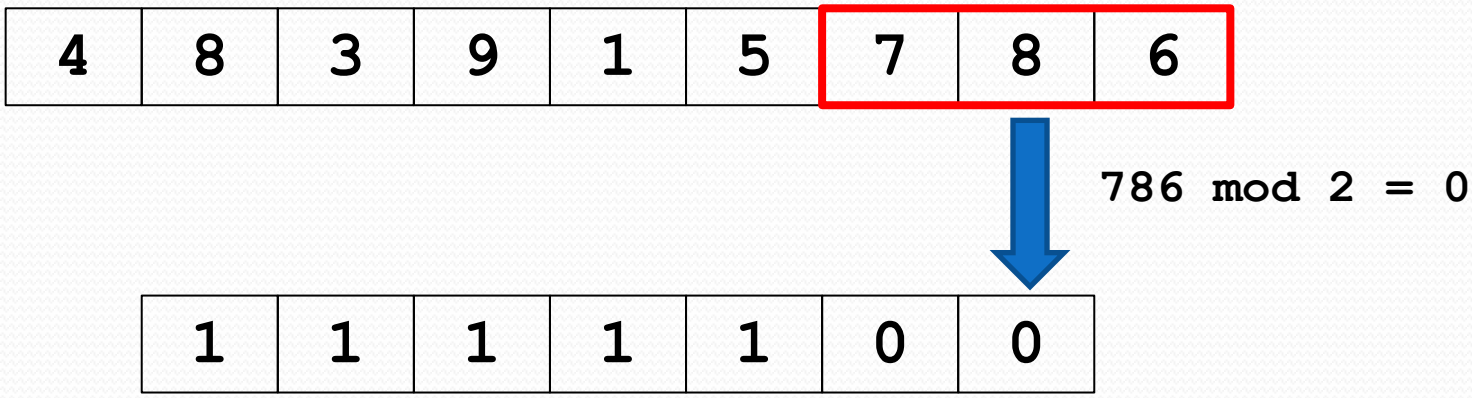


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:



Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:

4	8	3	9	1	5	7	8	6

1	1	1	1	1	0	0

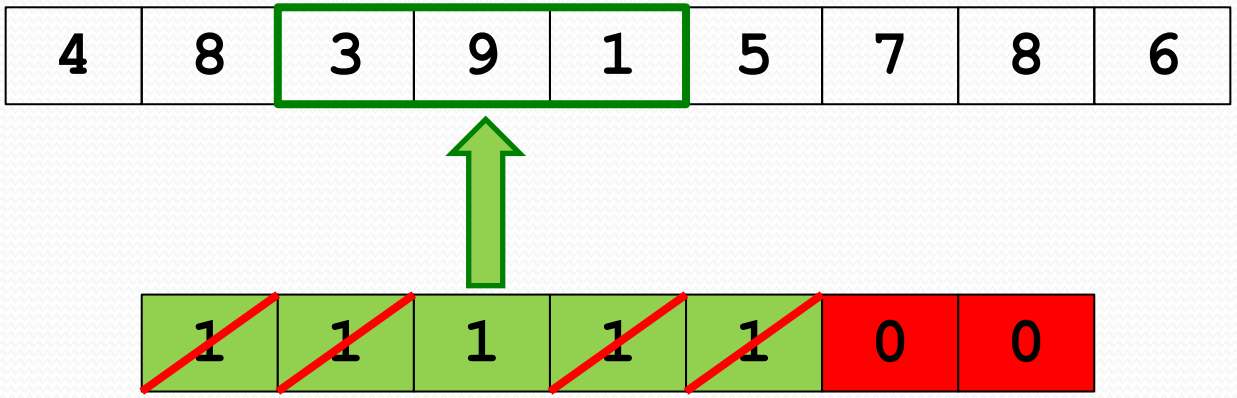
A zölddel jelölt helyeken egyezett meg a maradék a mintáéval.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 2, akkor az 1-es maradékot kell figyelni:



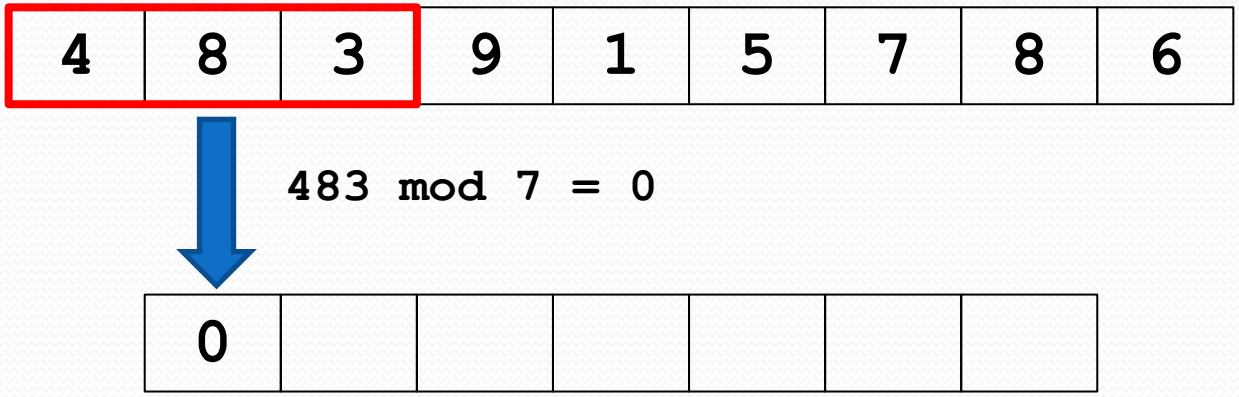
Észrevétel: többször is megkaptuk az 1-est maradékként, pedig csak egyetlen valódi egyezés volt; a többi „hamis” találat.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

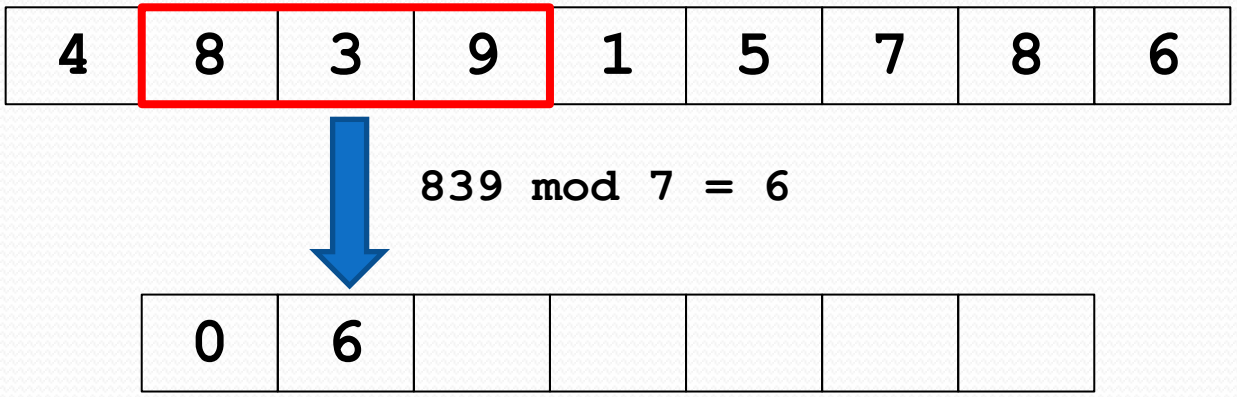


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

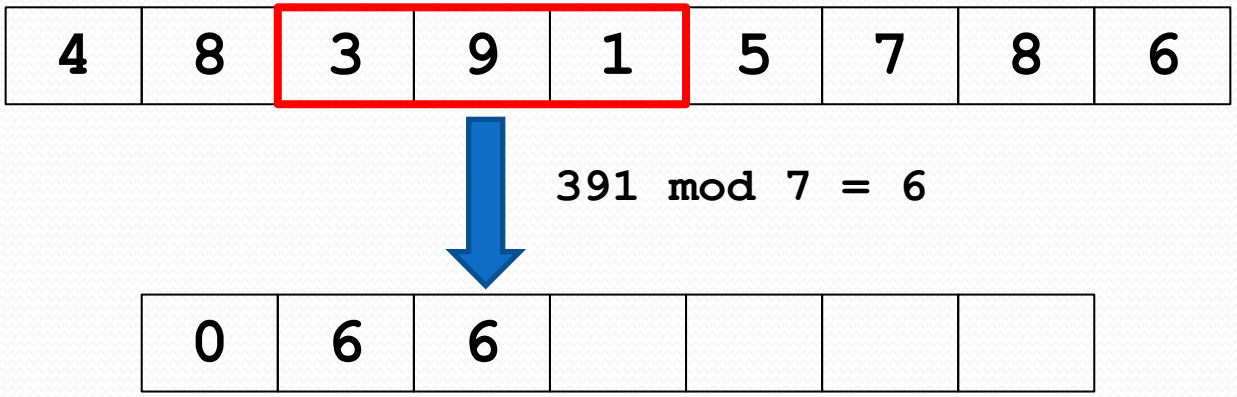


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

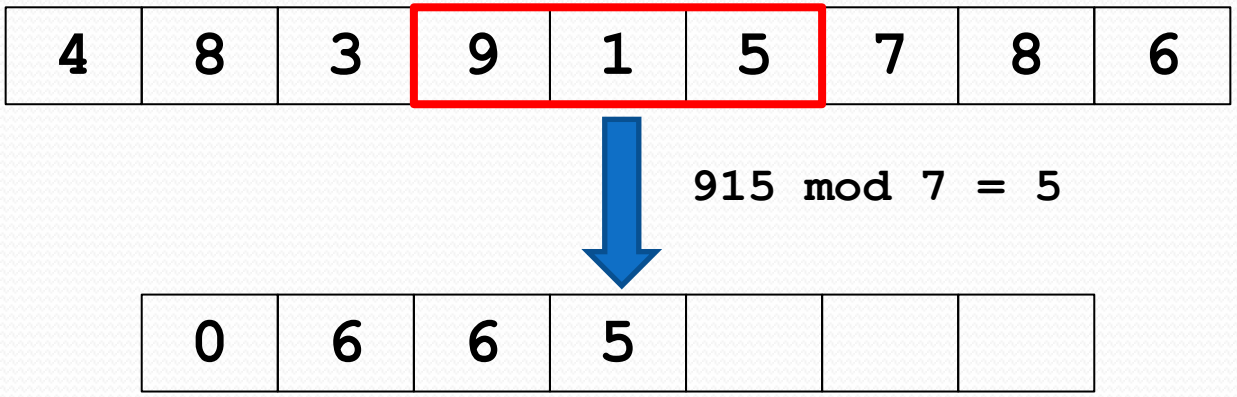


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

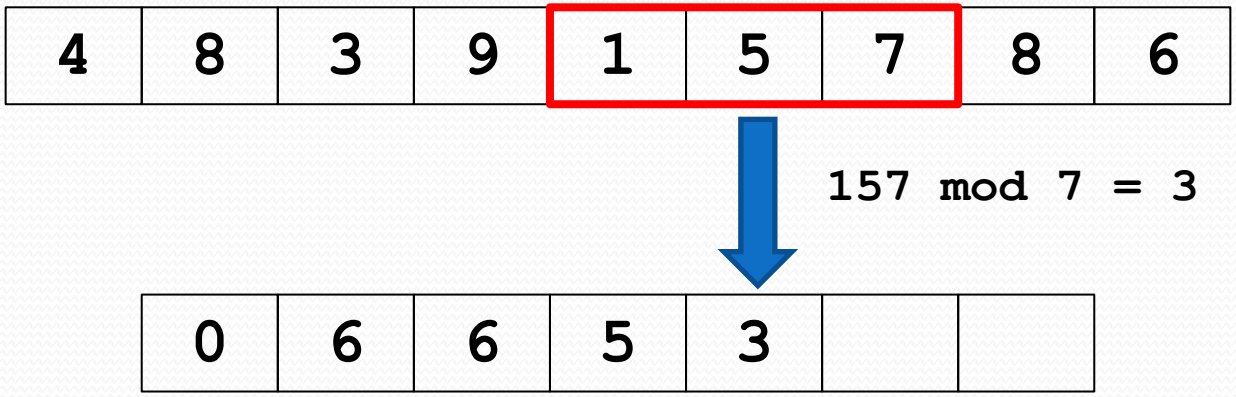


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

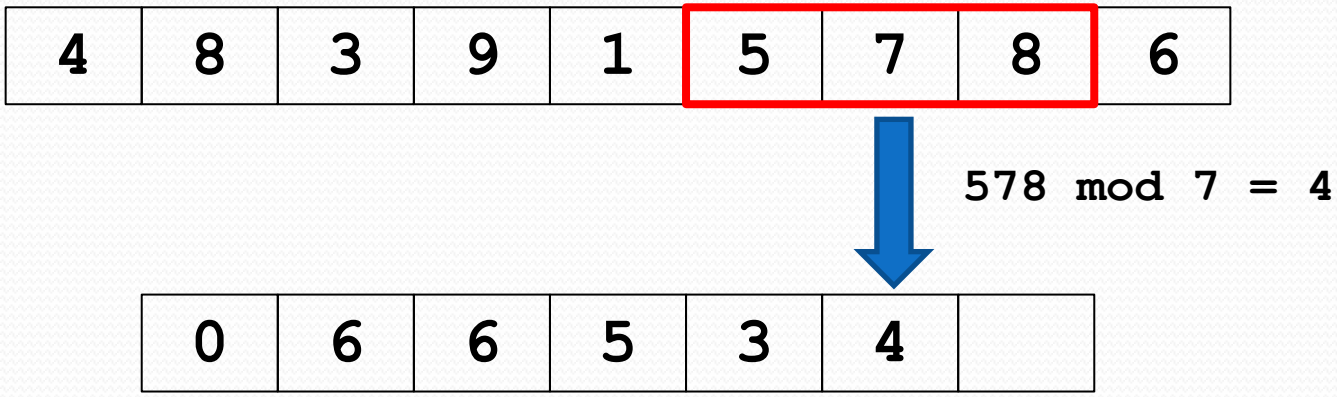


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

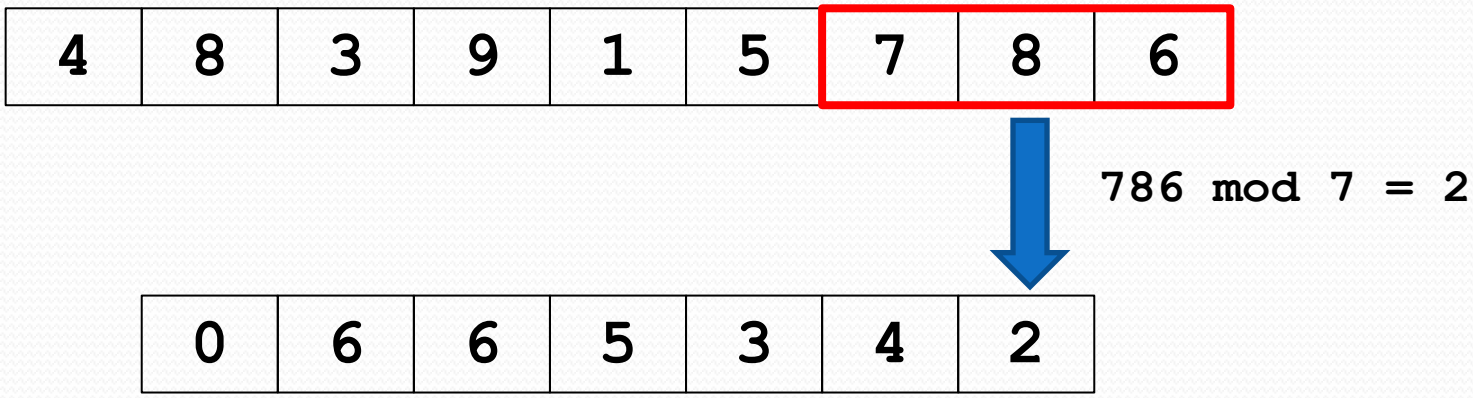


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:



Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839157 86**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:

4	8	3	9	1	5	7	8	6

0	6	6	5	3	4	2

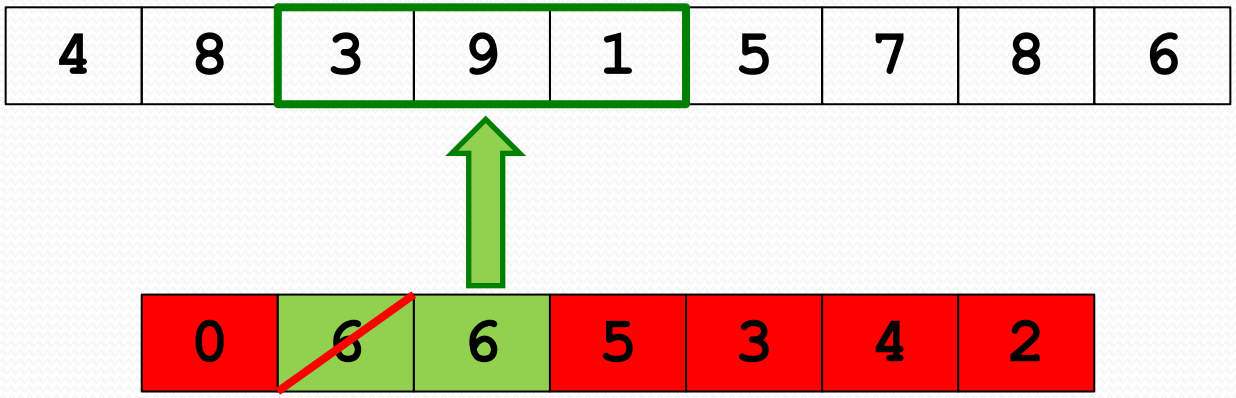
A zölddel jelölt helyeken egyezett meg a maradék a mintáéval.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **48391 5786**

Keresett minta: **391**

Ha a modulus 7, akkor a 6-os maradékot kell figyelni:



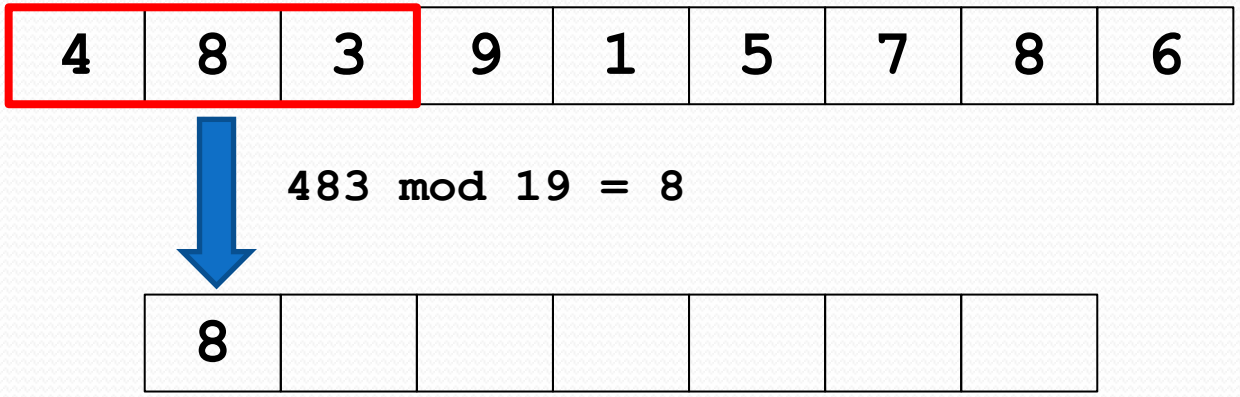
Észrevétel: többször is megkaptuk a 6-ost maradékként, pedig csak egyetlen valódi egyezés volt; a másik 6-os „hamis” találat.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

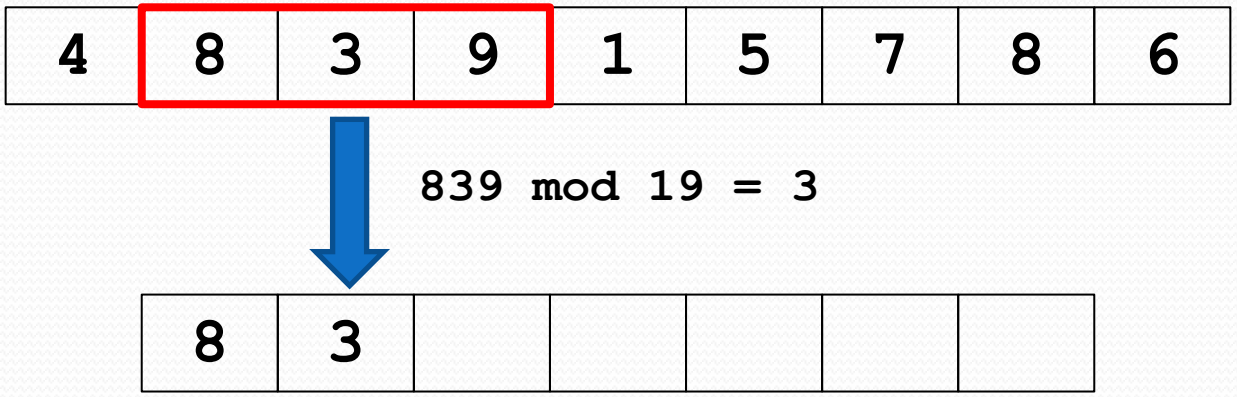


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

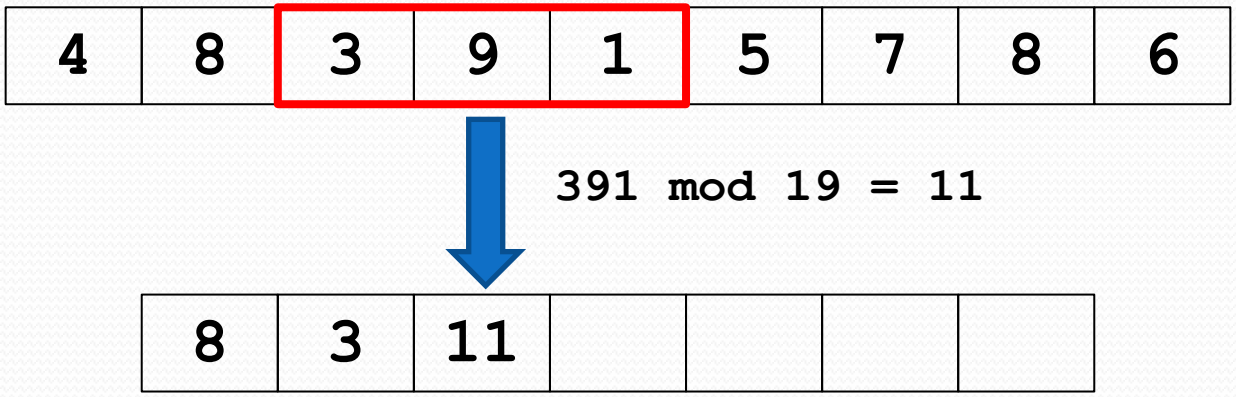


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

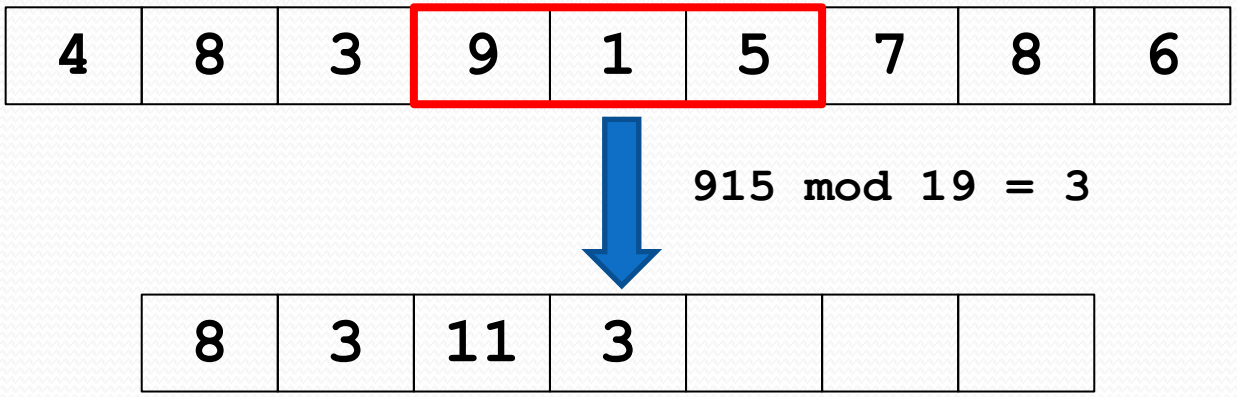


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839**15786

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

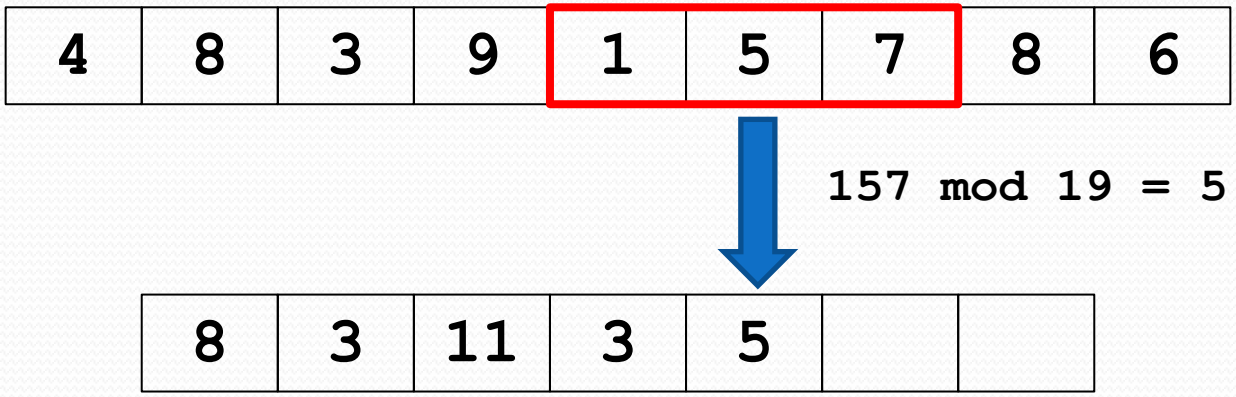


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

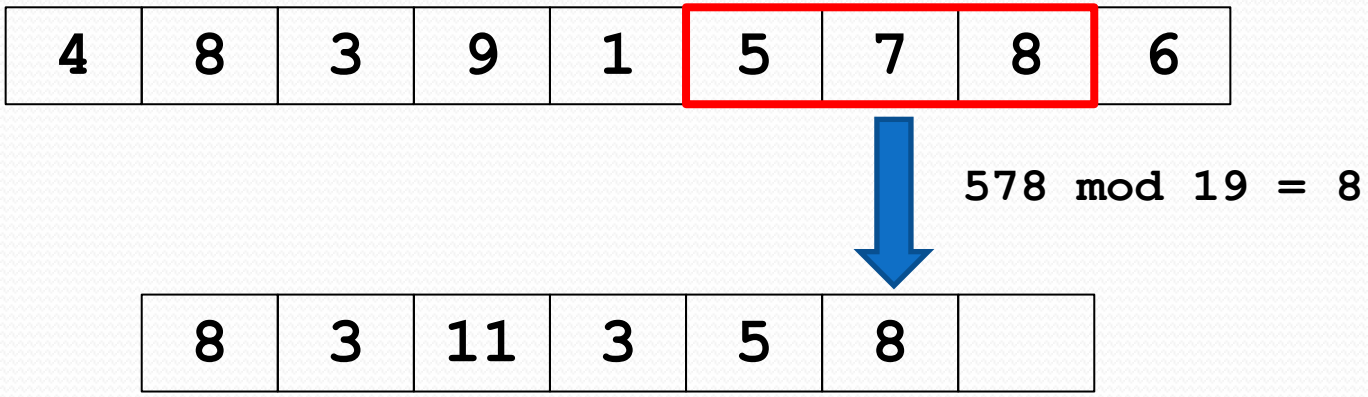


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

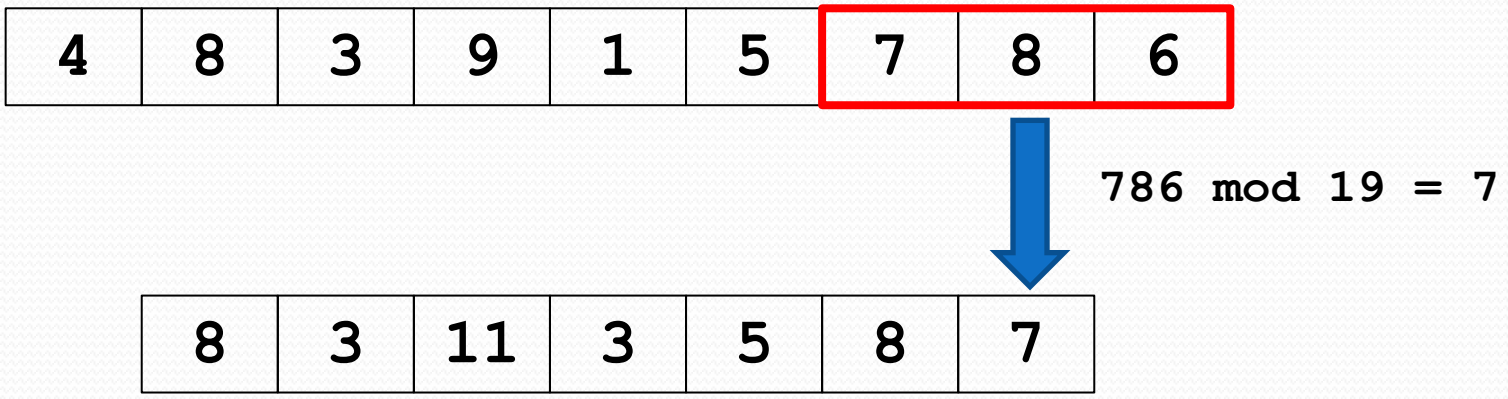


Az ötlet szemléltetése (folyt.)

Teljes sorozat: **483915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:



Az ötlet szemléltetése (folyt.)

Teljes sorozat: **4839915786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

4	8	3	9	1	5	7	8	6

8	3	11	3	5	8	7

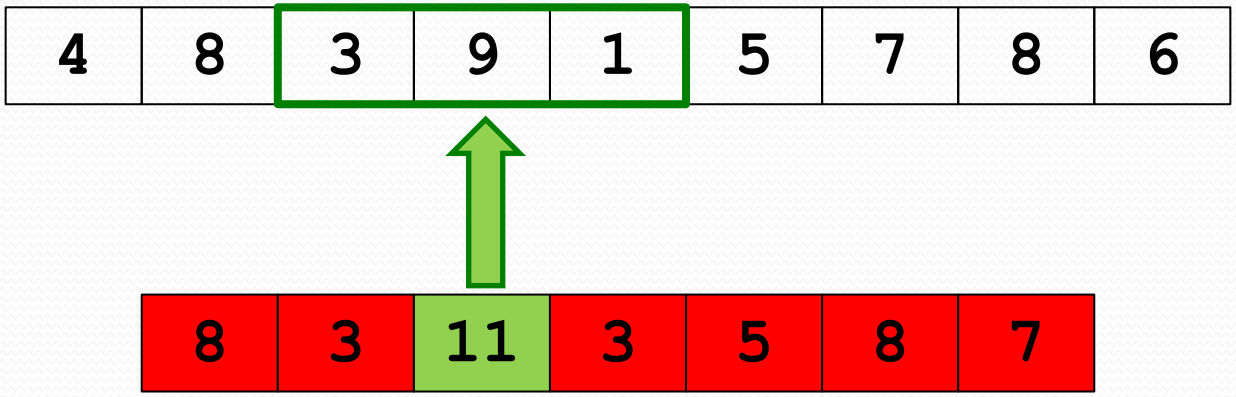
A zölddel jelölt helyen egyezett meg a maradék a mintáéval.

Az ötlet szemléltetése (folyt.)

Teljes sorozat: **48391 5786**

Keresett minta: **391**

Ha a modulus 19, akkor a 11-es maradékot kell figyelni:

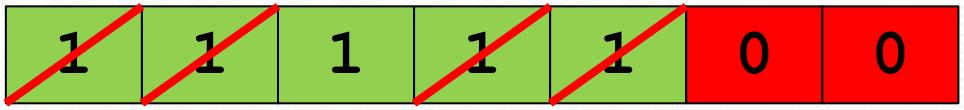


Észrevétel: csak egyszer kaptuk meg a 11-est maradékként; „hamis” találat nem volt (más számokkal ugyan lehetett volna).

Eredmények összevetése

Az ábrán jól látszik az egyező maradékok megoszlása, és levonható a következtetés: minél nagyobb modulust választunk, annál kisebb a valószínűsége egy „hamis” találatnak.

mod 2-nél:



mod 7-nél:

mod 19-nél:

Modulus megválasztása

A modulus megválasztása – vagyis hogy mely számmal vett osztási maradékot vesszük – fontos. Az előbbiekben láthattuk ennek okát.

Célunk tehát, hogy a hamis találatok számát minimalizáljuk, ehhez pedig a lehető legnagyobb modulus megválasztására van szükség.

Kérdés, hogy milyen nagy számot értünk ez alatt...

Modulus megválasztása (folyt.)

A Rabin-Karp algoritmus egy q prímszámot vezet be modulusként, melyet úgy szokás megválasztani, hogy $d \cdot q$ még éppen ábrázolható legyen az adott rendszeren, ahol d a számrendszer alapja.

Például 10-es számrendszer és 32 bites előjel nélküli egész esetén q ideális (legnagyobb) értéke $429\,496\,709$, mert $d \cdot q = 10 \cdot 429\,496\,709 = 4\,294\,967\,090$ (ez mindössze 205-tel kisebb (2^{32}-1)-nél).

(A rá következő prímszám 10-szeresét már nem tudnánk 32 biten ábrázolni.)

Újabb probléma

Ha a minta olyan nagy, hogy nem lehet ábrázolni, akkor az osztási maradékot sem tudjuk kiszámolni.

Némi trükkel megkerülhető ez a probléma, ugyanis számjegyenként haladva is meg lehet adni egy szám osztási maradékát. Az eljárást – az előbbi példától független mintával – a következő diák szemléltetik.

Osztási maradék meghatározása

$9816 \; mod \; 19 \; = ?$

Vezessük be az alábbi jelöléseket:

```
P:="9816"     //a minta (P az angol „pattern” után)
m:=hossz[P]   //a minta hossza (jelenleg 4)
d:=10         //a számrendszer alapja
q:=19         //választott prímszám
p:=0          //maradék (kezdetben 0)
```

Eljárás:

```
for i:=1 to m do
   p:=(d*p + P[i]) mod q
```

P=”**9816**”, d=10, q=19, p=**0**

p:=(d*p + P[i]) **mod** q

i=1 p:=(10*0 + **9**) **mod** 19 = 9 **mod** 19 = 9

P=”**9816**”, d=10, q=19, p=9

p:=(d*p + P[i]) **mod** q

i=1 p:=(10*0 + 9) **mod** 19 = 9 **mod** 19 = **9**

i=2 p:=(10*9 + 8) **mod** 19 = 98 **mod** 19 = 3

P=”**9816**”, d=10, q=19, p=3

p:=(d*p + P[i]) **mod** q

i=1 p:=(10*0 + 9) **mod** 19 = 9 **mod** 19 = 9

i=2 p:=(10*9 + 8) **mod** 19 = 98 **mod** 19 = **3**

i=3 p:=(10*3 + 1) **mod** 19 = 31 **mod** 19 = 12

P=”**9816**”, d=10, q=19, p=12

p:=(d*p + P[i]) **mod** q

i=1 p:=(10*0 + 9) **mod** 19 = 9 **mod** 19 = 9

i=2 p:=(10*9 + 8) **mod** 19 = 98 **mod** 19 = 3

i=3 p:=(10*3 + 1) **mod** 19 = 31 **mod** 19 = **12**

i=4 p:=(10***12**+ **6**) **mod** 19 =126 **mod** 19 = 12

P=”**9816**”, d=10, q=19, p=12

p:=(d*p + P[i]) **mod** q

i=1 p:=(10*0 + 9) **mod** 19 = 9 **mod** 19 = 9

i=2 p:=(10*9 + 8) **mod** 19 = 98 **mod** 19 = 3

i=3 p:=(10*3 + 1) **mod** 19 = 31 **mod** 19 = 12

i=4 p:=(10*12+ 6) **mod** 19 =126 **mod** 19 = **12**

Ellenőrzésképpen: 9816:19=516
31
126
12

Ezzel a módszerrel tehát számjegyenként haladva megállapítható az osztási maradék.

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]
 2. m:=hossz[P]
 3. h:=d^(m-1) mod q
 4. p:=0
 5. t_0:=0
 6. for i:=1 to m do begin
 7.    p:=(d*p + P[i]) mod q
 8.    t_0:=(d*t_0 + T[i]) mod q
 9. end

10. for s:=0 to n-m do begin
11.    if p=t_s then
12.       if P[1..m] = T[s+1..s+m] then
13.          print "A minta illeszkedik a(z) "(s+1)".
                    pozícióra"
14.    if s < n-m then
15.       t_s+1:=(d*(t_s - T[s+1]*h) + T[s+m+1]) mod q
16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]
 2. m:=hossz[P]
 3. h:=d^(m-1) mod q
 4. p:=0
 5. t_0:=0
 6. for i:=1 to m do begin
 7.    p:=(d*p + P[i]) mod q
 8.    t_0:=(d*t_0 + T[i]) mod q
 9. end

10. for s:=0 to n-m do begin
11.    if p=t_s then
12.       if P[1..m] = T[s+1..s+m] then
13.          print "A minta illeszkedik a(z) "(s+1)".
                    pozícióra"
14.    if s < n-m then
15.       t_s+1:=(d*(t_s - T[s+1]*h) + T[s+m+1]) mod q
16. end
```

A következőkben végignézzük az algoritmust lépésenként...

A Rabin-Karp algoritmus

`Rabin-Karp-illesztő(T, P, d, q)`

Az algoritmus bemenő paraméterei:

- `T`: teljes számsorozat (amiben keresünk)
- `P`: minta (amit keresünk T-ben)
- `d`: a számrendszer alapja ($2 \leq d \leq 10$)
- `q`: egy előre megválasztott tetszőleges prímszám

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]   //a teljes számsorozat (T) hossza
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
```

Magyarázat a következő dián...

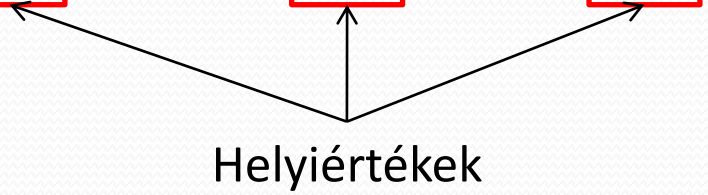
h:=d^(m-1) **mod** q magyarázata (előző diához)

A h változó tárolja egy m-jegyű szám legnagyobb helyiértékének q-val vett osztási maradékát.

Példaként nézzük az eddigi mintát:

391 = **3***100 + **9***10 + **1***1

= **3***10^2 + **9***10^1 + **1***10^0



Helyiértékek

A fenti példán látszik, hogy egy 3-jegyű szám legnagyobb helyiértéke (10-es számrendszer esetén) $10^{3-1} = 10^2$.

d alapú számrendszer és m-jegyű szám esetén tehát d^{m-1}.

$h := d^{m-1} \bmod q$ magyarázata (folyt.)

A h változó tárolja egy m-jegyű szám legnagyobb helyiértékének q-val vett osztási maradékát.

Tehát a $10^{3-1}=10^2=100$ helyiértéket még maradékosan el kell osztani q-val, amit most válasszunk meg 13-nak.

100:13=?

$h:=d^{m-1}$ **mod** q magyarázata (folyt.)

A h változó tárolja egy m-jegyű szám legnagyobb helyiértékének q-val vett osztási maradékát.

Tehát a $10^{3-1}=10^2=100$ helyiértéket még maradékosan el kell osztani q-val, amit most válasszunk meg 13-nak.

$100':13=7$ $\qquad$ $100=7*13+9$

9 ➡ példánkban tehát $h = 9$

(Erre a h értékre még szükség lesz az algoritmus során.)

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értékét fogja majd tárolni
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értékét fogja majd tárolni
 5. t_0:=0
```

Magyarázat a következő dián...

t_0 magyarázata (előző diához)

Az algoritmus során egy t_s változóban fogjuk tárolni, hogy az eredeti T sorozatban az $(s+1)$-edik pozíciótól számított mintányi (m) hosszúságú számnak mennyi a q-val vett osztási maradéka. p-hez hasonlóan t_s-t is számjegyenként haladva határozzuk majd meg. Az s 0-tól $(n-m)$-ig vesz fel értékeket.

(Esetünkben t_0 majd a (0+1)-edik pozíciótól vett 3-jegyű szám 13-mal vett osztási maradékát fogja tárolni.)

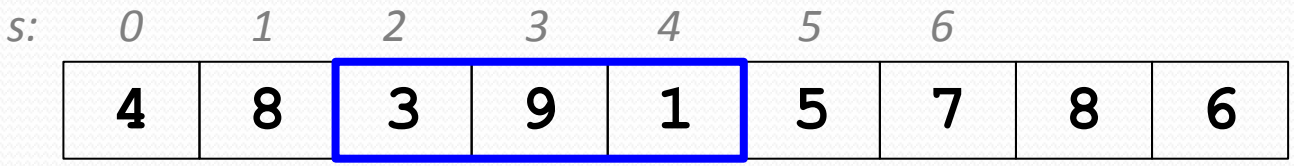
s:	0	1	2	3	4	5	6		
	4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

$$t_0 = 483 \bmod 13 = 2$$

(t_0 az 1. pozíciónál van)

t_0 magyarázata (előző diához)

Az algoritmus során egy t_s változóban fogjuk tárolni, hogy az eredeti T sorozatban az $(s+1)$-edik pozíciótól számított mintányi (m) hosszúságú számnak mennyi a q-val vett osztási maradéka. p-hez hasonlóan t_s-t is számjegyenként haladva határozzuk majd meg. Az s 0-tól $(n-m)$-ig vesz fel értékeket.



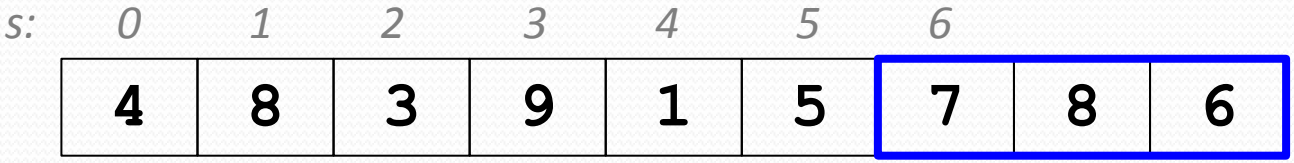
$t_2 = 391 \bmod 13 = 1$

$p = 391 \bmod 13 = 1$

(Majd ezt fogjuk keresni.)

t_0 magyarázata (előző diához)

Az algoritmus során egy t_s változóban fogjuk tárolni, hogy az eredeti T sorozatban az $(s+1)$-edik pozíciótól számított mintányi (m) hosszúságú számnak mennyi a q-val vett osztási maradéka. p-hez hasonlóan t_s-t is számjegyenként haladva határozzuk majd meg. Az s 0-tól $(n\text{-}m)$-ig vesz fel értékeket.



$t_6 = 786 \bmod 13 = 6$

($n\text{-}m$ = 9-3 = 6 az utolsó pozíció)

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás


 9. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
                             (számjegyenként)
 9. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t_0:=(d*t_0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                (számjegyenként)
```

Megjegyzés: $T[1..m]$ a T karaktersorozat 1.-től m-edik karakteréig tartó részt jelenti.

A for-ciklus befejeztével megvan p és t_0 értéke (p már nem fog változni az algoritmus során).

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t_0:=(d*t_0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                (számjegyenként)
10. for s:=0 to n-m do begin //illesztés
16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t_0:=(d*t_0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                  (számjegyenként)

10. for s:=0 to n-m do begin //illesztés
11.    if p=t_s then //osztási maradék megegyezik?




16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t_0:=(d*t_0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                  (számjegyenként)

10. for s:=0 to n-m do begin //illesztés
11.    if p=t_s then //osztási maradék megegyezik?
12.       if P[1..m] = T[s+1..s+m] then //minta illeszkedik?




16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t0:=(d*t0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                  (számjegyenként)

10. for s:=0 to n-m do begin //illesztés
11.    if p=ts then //osztási maradék megegyezik?
12.       if P[1..m] = T[s+1..s+m] then //minta illeszkedik?
13.          print "A minta illeszkedik a(z) "(s+1)".
                    pozícióra"



16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t0:=(d*t0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                  (számjegyenként)

10. for s:=0 to n-m do begin //illesztés
11.    if p=ts then //osztási maradék megegyezik?
12.       if P[1..m] = T[s+1..s+m] then //minta illeszkedik?
13.          print "A minta illeszkedik a(z) "(s+1)".
                    pozícióra"
14.    if s < n-m then //ha nem értünk T végére

16. end
```

A Rabin-Karp algoritmus

```
Rabin-Karp-illesztő(T, P, d, q)
 1. n:=hossz[T]    //a teljes számsorozat (T) hossza
 2. m:=hossz[P]    //a minta hossza
 3. h:=d^(m-1) mod q
 4. p:=0           //P mod q értéke lesz majd
 5. t_0:=0
 6. for i:=1 to m do begin //előfeldolgozás
 7.    p:=(d*p + P[i]) mod q //P mod q meghatározása
 8.    t_0:=(d*t_0 + T[i]) mod q //T[1..m] mod q meghatározása
 9. end                                  (számjegyenként)

10. for s:=0 to n-m do begin //illesztés
11.    if p=t_s then //osztási maradék megegyezik?
12.       if P[1..m] = T[s+1..s+m] then //minta illeszkedik?
13.          print "A minta illeszkedik a(z) "(s+1)".
                    pozícióra"
14.    if s < n-m then //ha nem értünk T végére
15.       t_s+1:=(d*(t_s - T[s+1]*h) + T[s+m+1]) mod q
16. end
```

Magyarázat a következő dián...

t_{s+1} magyarázata (előző diához)

Ha a mintaillesztési folyamat során nem találtunk egyezést, akkor eggyel jobbra kell léptetni az m szélességű ablakot, és kiszámolni az új maradékot; erre szolgál az alábbi képlet:

$$t_{s+1} := (d*(t_s - T[s+1]*h) + T[s+m+1]) \bmod q$$

Megértéséhez nézzük először a képletet maradékos osztások nélkül:

Mivel $h = d^{m-1} \bmod q$, ezért maradékos osztás nélkül csak d^{m-1}.

$$t_{s+1} := d*(t_s - T[s+1]*d^{m-1}) + T[s+m+1]$$

Ablaklépteté si példa

Legyen a számsorozat a 9816, az ablakméret pedig 3.

Ebből következik, hogy az ablak kezdetben a 981-et fogja tartalmazni. Ha eggyel jobbra szeretnénk csúsztatni az ablakot, akkor ebből a 981-ből 816-ot kellene kapnunk.

Nagyvonalakban ugyebár a 9-est „eltüntetjük”, a 6-ost pedig az ott maradt 81 mögé „írjuk”. Kérdés, hogy ezt matematikailag hogyan lehet kivitelezni...

Ablakléptetési példa (folyt.)

Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.

981 **9816**

t_{s+1}-et mindig t_s-ből számoljuk, tehát az biztosan szerepel a képletben.

$t_{s+1} := t_s$

Ablakléptetési példa (folyt.)

Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.

9816

981

−900

Levonjunk a legnagyobb helyiértéken lévő számot (a megfelelő helyiértékkel beszorozva).
Esetünkben $T[s+1] = 9$, valamint $d^{m-1} = 10^{3-1} = 10^2 = 100$.

$t_{s+1} := t_s - T[s+1] * d^{m-1}$

Ablakléptetési példa (folyt.)

Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.

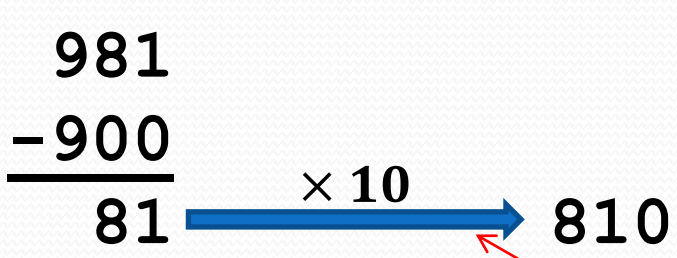
```
  981
 -900
 ----
   81
```

9816

$t_{s+1} := t_s - T[s+1] * d^{m-1}$

Ablakléptetési példa (folyt.)

Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.

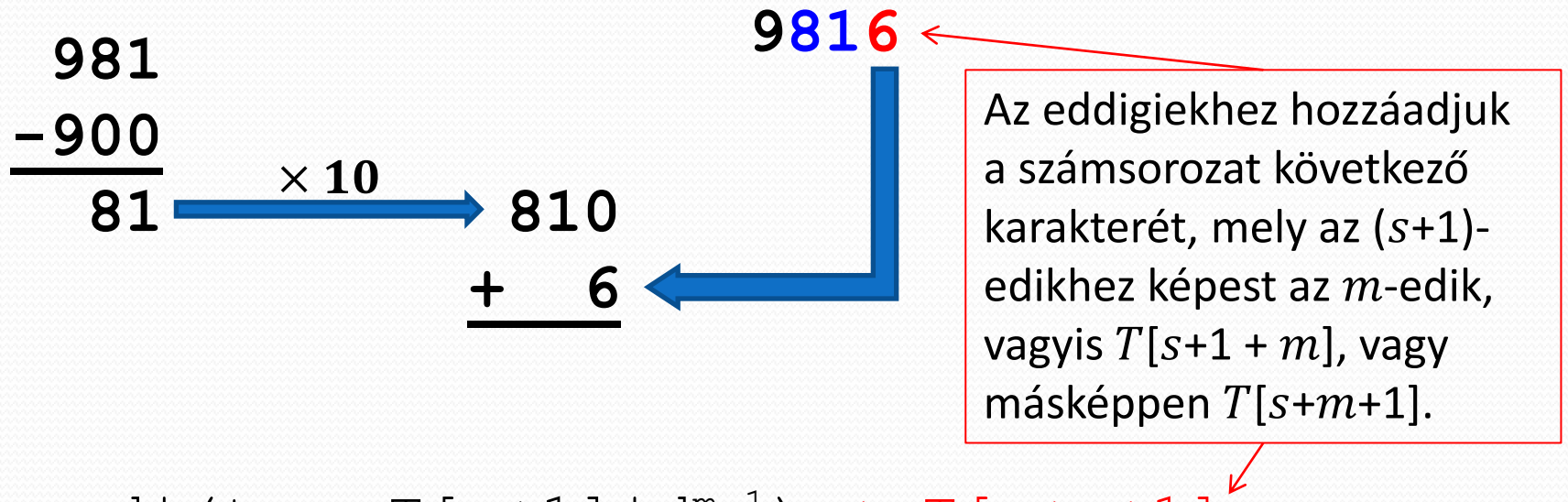


Beszorzunk a számrendszer alapjával, vagyis d-vel.

$$t_{s+1} := d * (t_s - T[s+1] * d^{m-1})$$

Ablakléptetési példa (folyt.)

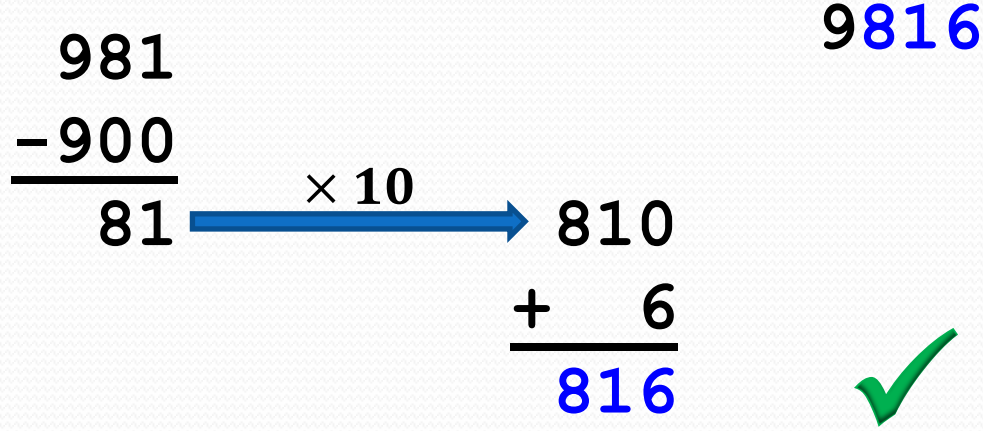
Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.



$t_{s+1} := d*(t_s - T[s+1]*d^{m-1}) + T[s+m+1]$

Ablakléptetési példa (folyt.)

Átlagos esetben azt mondhatjuk, hogy a legnagyobb helyiértéken lévő számot (a helyiértékkel szorozva) kivonjuk, a számsorozat következő számjegyét pedig hozzáadjuk az így kapott szám d-szereséhez.



$t_{s+1} := d*(t_s - T[s+1]*d^{m-1}) + T[s+m+1]$

A helyes képlet

Ne feledjük azonban, hogy az algoritmus szerint a t_s osztási maradékot kell tároljon, szóval a **mod** q-t nem árt visszaírni:

$t_{s+1} := d*(t_s - T[s+1]*d^{m-1}) + T[s+m+1]$

$t_{s+1} := (d*(t_s - T[s+1]*(d^{m-1} \textbf{ mod } q)) + T[s+m+1]) \textbf{ mod } q$

$t_{s+1} := (d*(t_s - T[s+1]*h) + T[s+m+1]) \textbf{ mod } q$

Példa levezetése

Legyen adva a korábbi számsorozat és minta:

Teljes sorozat: **4839157 86**

Keresett minta: **391**

Továbbá a könnyebb számolás végett hagyjuk meg a korábban választott prímszámot, vagyis a q=13-at.

A számrendszer alapja értelemszerűen d=10.

A megoldás során vörös szegélyű téglalapok fognak segíteni emlékeztetőkkel, tippekkel.

Példa levezetése

T=483915786 d=10

P=391 q=13

Példa levezetése

T=483915786 d=10 n=9 ← `n = hossz[T]`

P=391 q=13

Példa levezetése

T=483915786 d=10 n=9

P=391 q=13 m=3 ← m = hossz[P]

Példa levezetése

T=483915786 d=10 n=9

P=391 q=13 m=3

$h = 10^{3-1} \bmod 13 = 100 \bmod 13 = 9$

$h = d^{m-1} \bmod q$

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3

Helymegtakarítás végett átírtam ide h-t.

Példa levezetése (előfeldolgozás)

```
P=391   d=10
p=0
i=1 p=(10*0 + 3) mod 13 = 3 mod 13 = 3
```

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Tipp: Hogy megkíméljük magunkat sok fejszámolástól, írjuk fel a modulus – esetünkben a 13 – többszöröseit mondjuk 10-ig (szükség szerint lehet a listát bővíteni). Így rögtön látszik majd, hogy egy adott számban hányszor van meg egész számszor a 13, és már csak a különbséget kell fejben kiszámolni.
Például: 125 mod 13=? Ez esetben megkeressük a legnagyobb számot a listában, ami kisebb-egyenlő 125-tel – ez most a 117; a maradék pedig a kettő különbsége lesz: 125-117=8.
Tehát 125 mod 13=8.
A továbbiakban piros színnel ki lesznek húzva azon többszörösök, melyek kisebb-egyenlők a keresett számmal.

Példa levezetése (előfeldolgozás)

P=391 d=10

p=0

i=1 p=(10*0 + 3) **mod** 13 = 3 **mod** 13 = 3

i=2 p=(10*3 + 9) **mod** 13 =39 **mod** 13 = 0

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése (előfeldolgozás)

P=391 d=10

p=0

i=1 p=(10*0 + 3) **mod** 13 = 3 **mod** 13 = 3

i=2 p=(10*3 + 9) **mod** 13 =39 **mod** 13 = 0

i=3 p=(10*0 + 1) **mod** 13 = 1 **mod** 13 = 1

p=1

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése (előfeldolgozás)

P=391 d=10

p=0

i=1 p=(10*0 + 3) **mod** 13 = 3 **mod** 13 = 3

i=2 p=(10*3 + 9) **mod** 13 =39 **mod** 13 = 0

i=3 p=(10*0 + 1) **mod** 13 = 1 **mod** 13 = 1

p=1

t_0=0

i=1 p=(10*0 + 4) **mod** 13 = 4 **mod** 13 = 4

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése (előfeldolgozás)

P=391 d=10

p=0

i=1 p=(10*0 + 3) **mod** 13 = 3 **mod** 13 = 3

i=2 p=(10*3 + 9) **mod** 13 =39 **mod** 13 = 0

i=3 p=(10*0 + 1) **mod** 13 = 1 **mod** 13 = 1

p=1

t_0=0

i=1 p=(10*0 + 4) **mod** 13 = 4 **mod** 13 = 4

i=2 p=(10*4 + 8) **mod** 13 =48 **mod** 13 = 9

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése (előfeldolgozás)

P=391 d=10

p=0

i=1 p=(10*0 + 3) **mod** 13 = 3 **mod** 13 = 3

i=2 p=(10*3 + 9) **mod** 13 =39 **mod** 13 = 0

i=3 p=(10*0 + 1) **mod** 13 = 1 **mod** 13 = 1

p=1

t_0=0

i=1 p=(10*0 + 4) **mod** 13 = 4 **mod** 13 = 4

i=2 p=(10*4 + 8) **mod** 13 =48 **mod** 13 = 9

i=3 p=(10*9 + 3) **mod** 13 =93 **mod** 13 = 2

t_0=2

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

$t_0=2$

4	8	3	9	1	5	7	8	6

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

Sorszámozzuk be a karaktereket (magunknak lesz segítség).

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ← Az s 0-tól megy (n-m)-ig.

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0

p = t_0 ?

← Osztási maradék megegyezik?

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ?

p = t_0

1 = 2 ✗

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ?

p = t_0

1 = 2 ✗

s < n−m ? ← Még nem értünk a T számsorozat végére.

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_0=2

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ?

p = t_0

1 = 2 ✗

s < n-m ?

0 < 6 ✓

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

$t_0=2$

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ?

p = t_0

1 = 2 ✗

$t_{s+1}:=(d*(t_s-T[s+1]*h) + T[s+m+1])$ **mod** q

s < n-m ?

0 < 6 ⇒ $t_1=(10*(2 - 4*9) + 9) \bmod 13 = -331 \bmod 13$

✓

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

$t_0=2$

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=0 ?

p = t_0

1 = 2 ✗

s < n−m ?

0 < 6 → $t_1=(10*(2 - 4*9) + 9) \bmod 13 = -331 \bmod 13$

+390

59 mod 13 = 7

✓

Keressük meg 13-nak azt a többszörösét, ami nagyobb-egyenlő 331-nél, majd ezt adjuk hozzá a -331-hez, hogy visszalépjünk a pozitív számok körébe. Célszerű ilyenkor 13 többszöröseit felírni (lásd jobbra), és vagy ezek közt keresni, vagy az egyszerűség kedvéért pl. a 10-szeresüket venni. 331-nél például a 390 (39*10) biztosan nagyobb lesz, így ezt hozzáadva -331-hez 59-et kapunk.

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

$t_1=7$

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

1 = 7 ✘

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

1 = 7 ✗

s ? < n-m

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

1 = 7 ✗

s ? < n-m

1 < 6 ✓

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

1 = 7 ✗

t_{s+1}:=(d*(t_s-T[s+1]*h) + T[s+m+1]) **mod** q

s < n-m ?

1 < 6 ⇒ t_2=(10*(7 – 8*9) + 1) mod 13 = -649 mod 13

✓

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_1=7

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=1 ?

p = t_1

1 = 7 ✗

s < n-m ?

1 < 6 → t_2=(10*(7 − 8*9) + 1) mod 13 = −649 mod 13

+650

1 mod 13 = 1

0	0
1	13
2	26
3	39
4	52
5	65
6	78
7	91
8	104
9	117
10	130

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

$t_2=1$

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=2

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=2 ?

p = t_2

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=2 ?

p = t_2

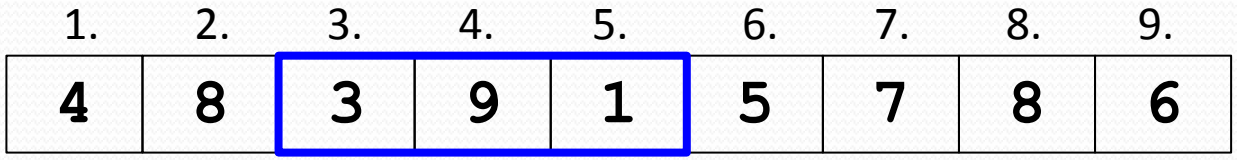
1 = 1

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1



s=2 ?

p = t_2

1 = 1 ✓

P[1..m] = T[s+1..s+m]

?

P[1..3] = T[3..5]

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=2

p = t_2 ?

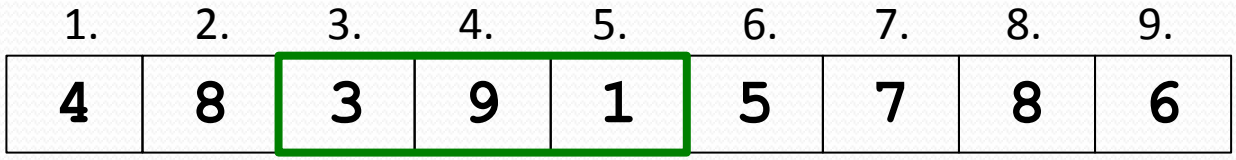
1 = 1 ⇒ P[1..3] ? = T[3..5] ✓

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1



s=2

? ?

p = t_2 P[1..3] = T[3..5]

1 = 1

s+1

”A minta illeszkedik a(z) 3. pozícióra.”

Példa levezetése

T=483915786 d=10 n=9 h=9

P=391 q=13 m=3 p=1

t_2=1

1.	2.	3.	4.	5.	6.	7.	8.	9.
4	**8**	**3**	**9**	**1**	**5**	**7**	**8**	**6**

s=2

$p \overset{?}{=} t_2$

$1 = 1$ → $P[1..3] \overset{?}{=} T[3..5]$ ✓

"A minta illeszkedik a(z) 3. pozícióra."

(A további vizsgálat opcionális attól függően, hogy csak első egyezésig megyünk, vagy az összes illeszkedést meg szeretnénk keresni. Az egyszerűség kedvéért most nem nézzük tovább a feladatot; az előzőekhez hasonlóan lehetne folytatni *s*=6-ig.)