Olgu antud mingi *tähestik* Σ — lihtsalt mingi lõplik hulk. Tema elemente nimetame *tähtedeks*.

Sõne on tähestiku elementide mingi (lõplik) järjend. Kõigi sõnede hulka üle tähestiku Σ tähistame Σ^*-ga.

Sõne t *pikkus* on tähtede arv temas. Tähistame $|t|$.

Sõne t tähti tähistame $t[1], t[2], \ldots, t[|t|]$. S.t. $t = t[1]t[2]\cdots t[|t|]$.

Alamsõne: $t[i \ldots j] := t[i]t[i+1]\cdots t[j]$.

Tühja sõnet — ainukest sõnet pikkusega 0 — tähistame ϵ-ga.

Kui $i > j$, siis $t[i \ldots j] := \epsilon$.

Olgu $s, t \in \Sigma^*$. s *esineb* t-s *positsioonis* i, kui $t[i \ldots i + |s| - 1] = s$.

s on t *prefiks*, kui s esineb t-s positsioonis 1. Tähistame $s \sqsubset t$.

s on t *sufiks*, kui s esineb t-s positsioonis $|t| - |s| + 1$. Tähistame $s \sqsupset t$.

Ülesanne: antud s ja t. Leia kõik sellised positsioonid i, et s esineb t-s positsioonis i.

Naiivne algoritm:

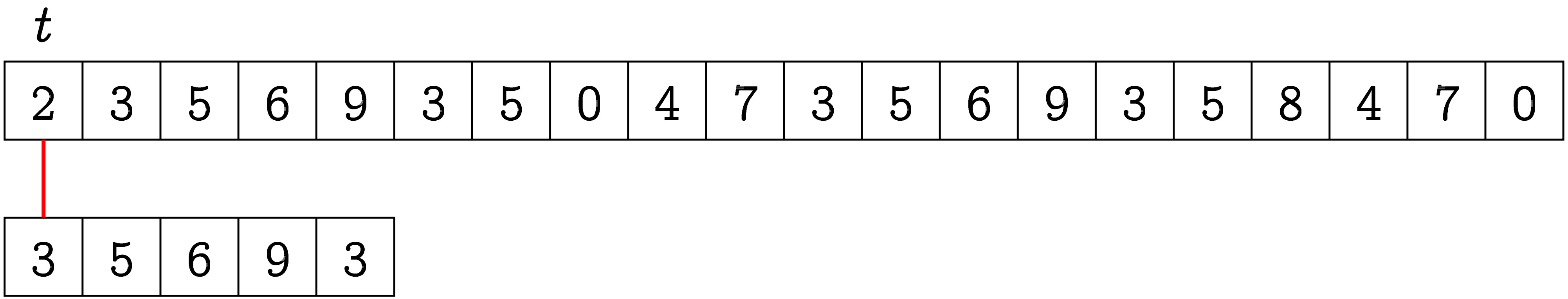
```
$J := \emptyset$; $m := |s|$; $n := |t|$
**for** $i := 1$ **to** $n - m + 1$ **do**
    **if** võrdle_sõnesid($s, m, t, i$) $= 0$ **then** $J \Leftarrow i$
**return** $J$
```

võrdle_sõnesid(s, m, t, i) kontrollib, kas sõne s pikkusega m on võrdne alamsõnega $t[i \ldots i+m-1]$. Tagastab vähima sellise j, kus $s[j] \neq t[i+j-1]$. Kui sellist ei leidu, siis tagastab 0. Ta on:

```
**for** $j := 1$ **to** $m$ **do**
    **if** $s[j] \neq t[i-1+j]$ **then**
        **return** $j$
**return** 0
```

Keerukus: $\Theta(m(n-m))$, kus $m = |s|$ ja $n = |t|$.

$i = 1$ $J = \emptyset$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

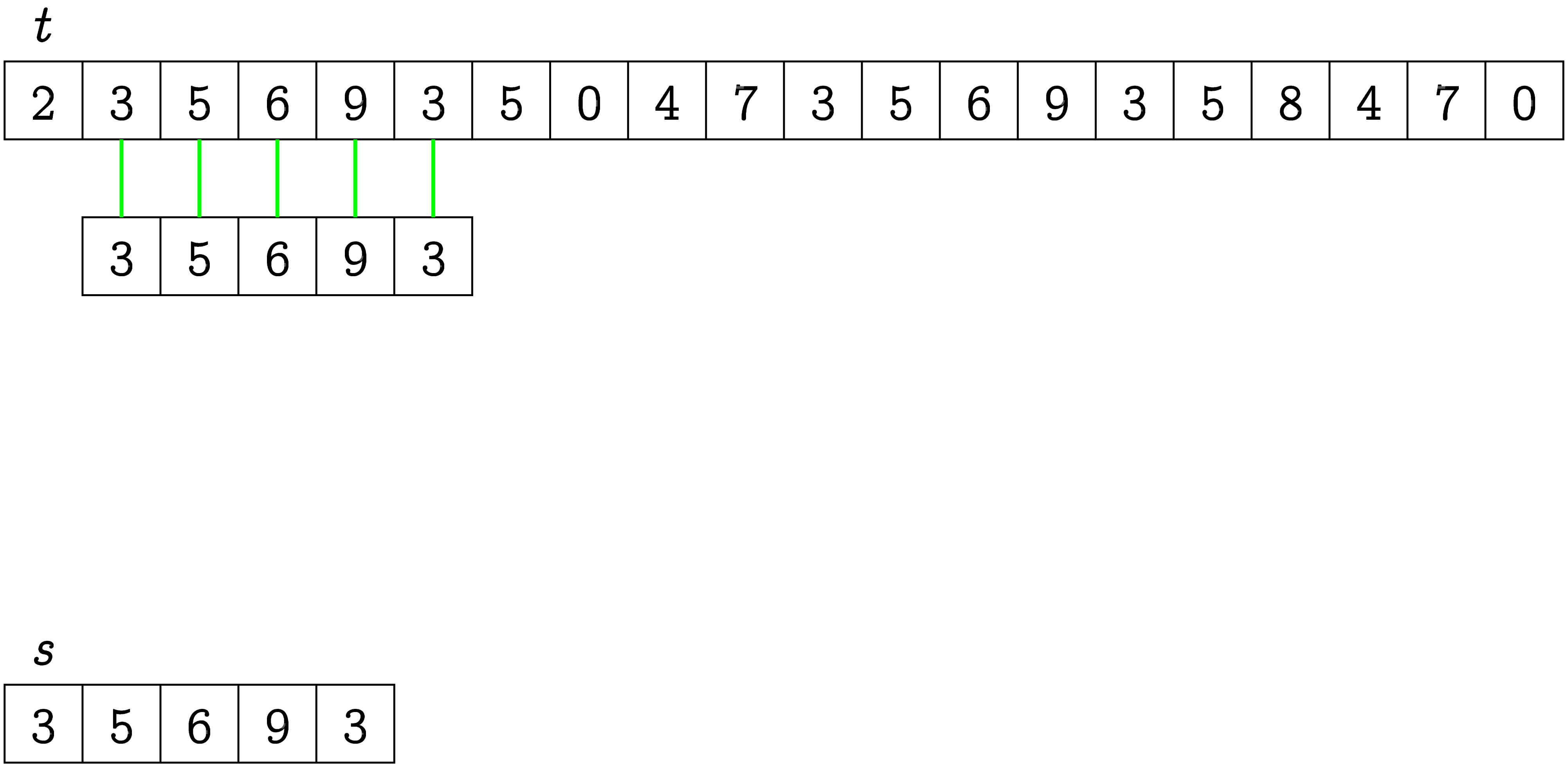
3	5	6	9	3

s

3	5	6	9	3

$i = 2$

$J = \{2\}$



t

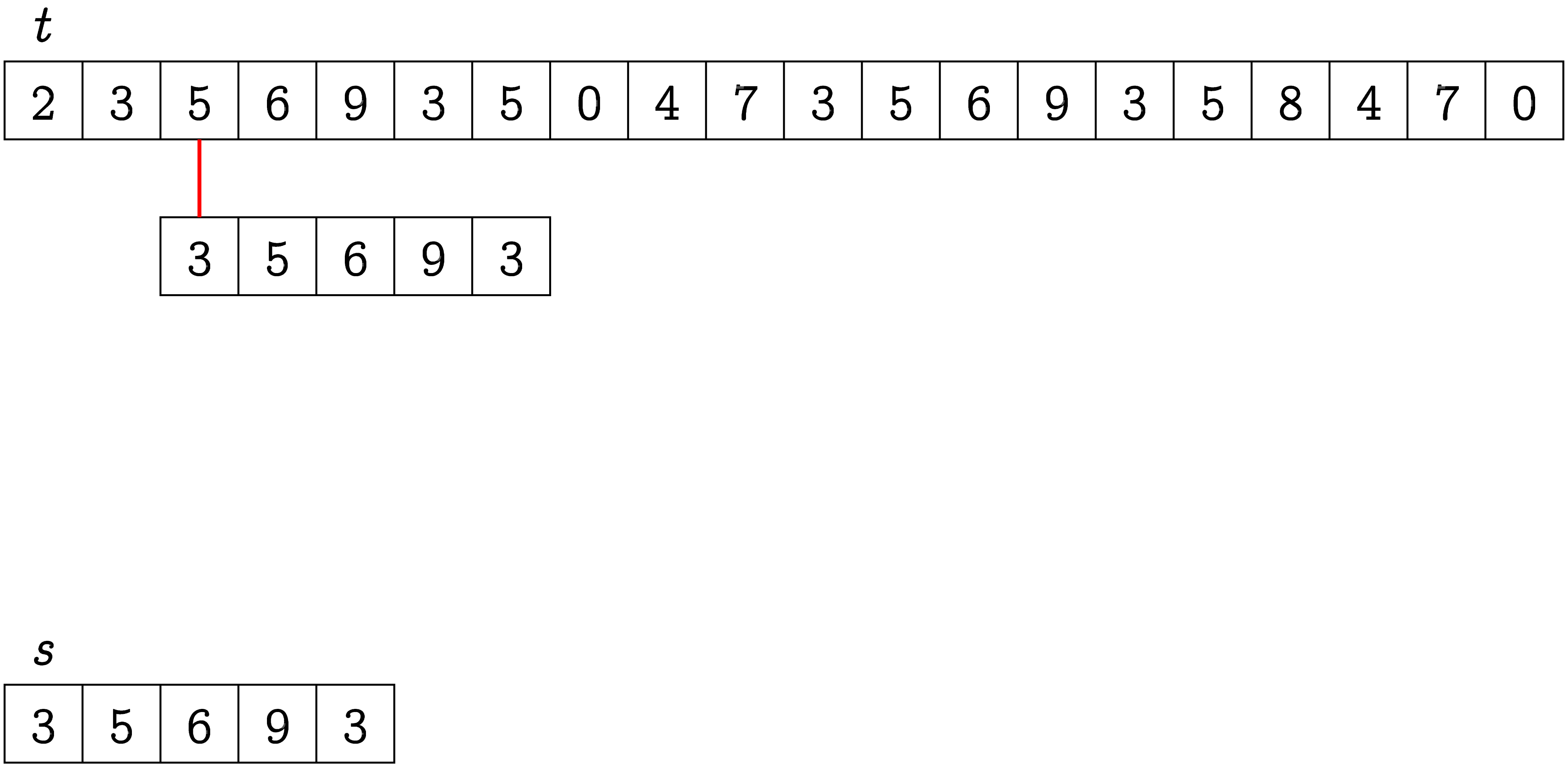
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 3$ $J = \{2\}$



t

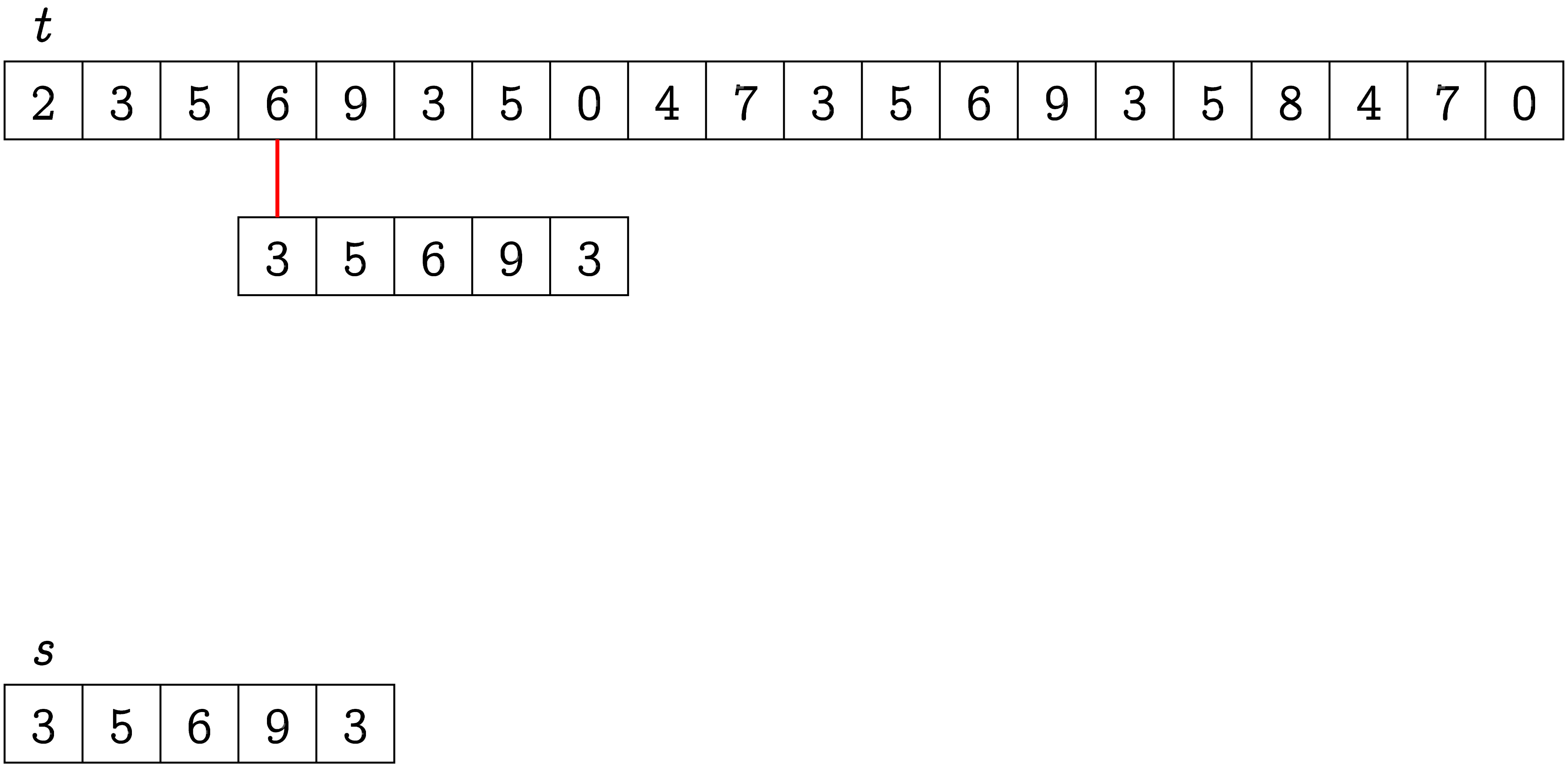
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

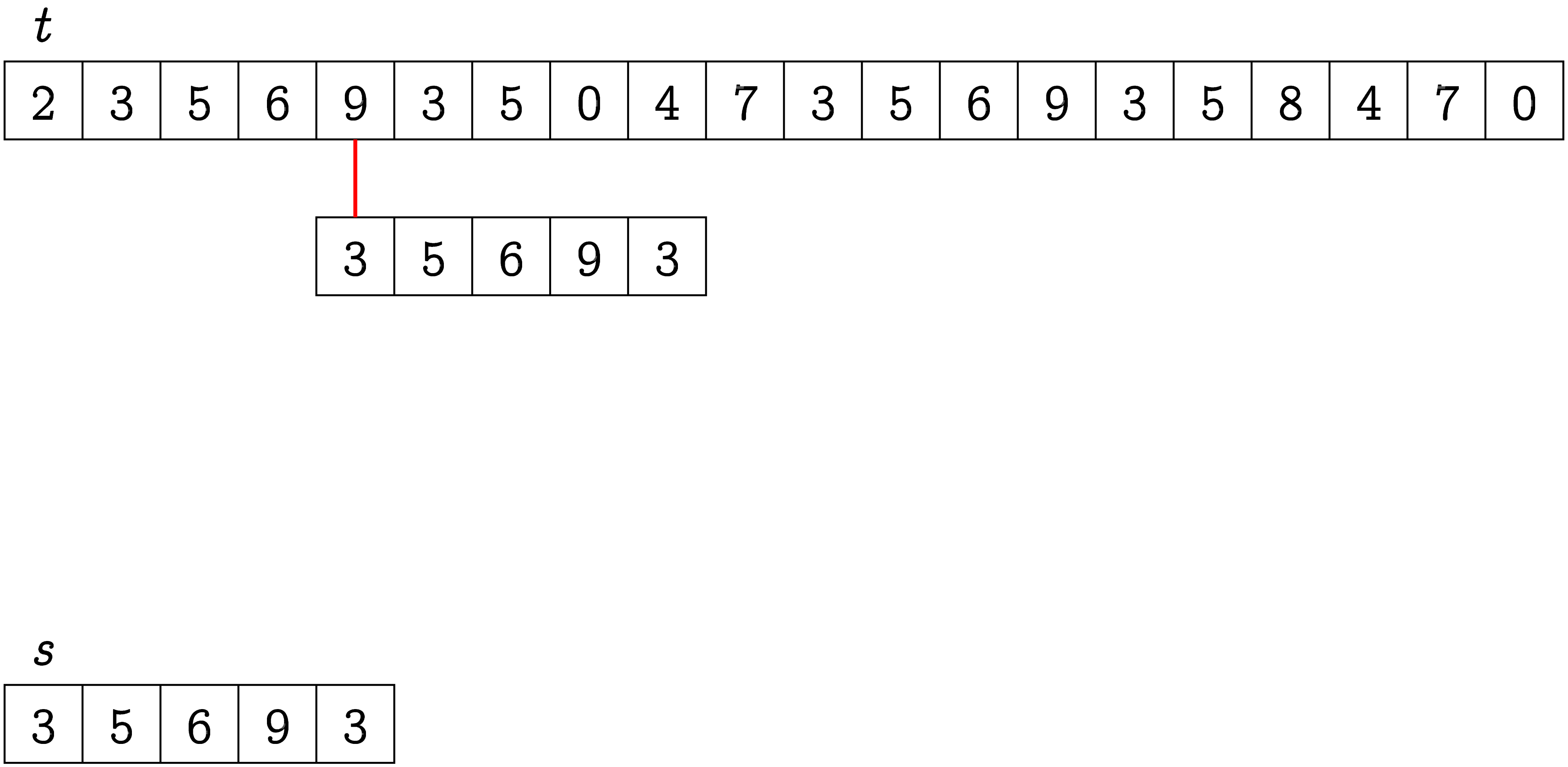
s

3	5	6	9	3

$i = 4$ $J = \{2\}$



$i = 5$ $J = \{2\}$



t

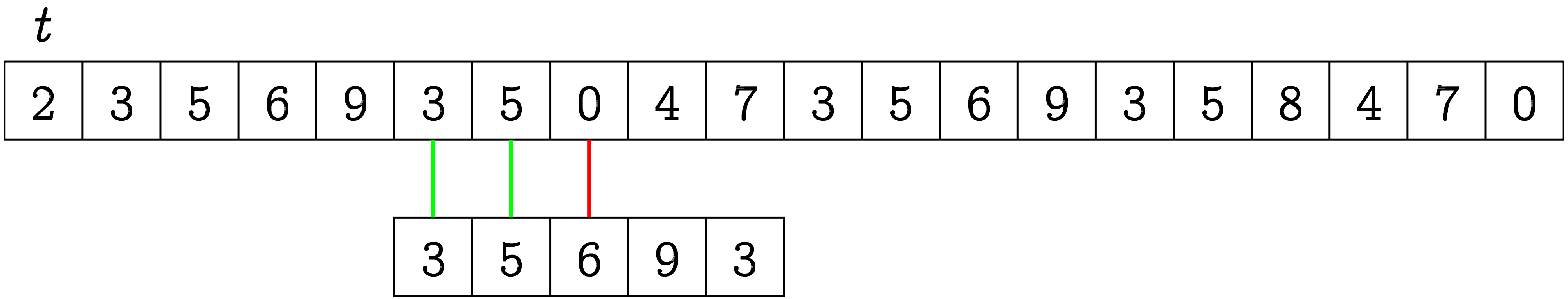
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 6$ $J = \{2\}$



t

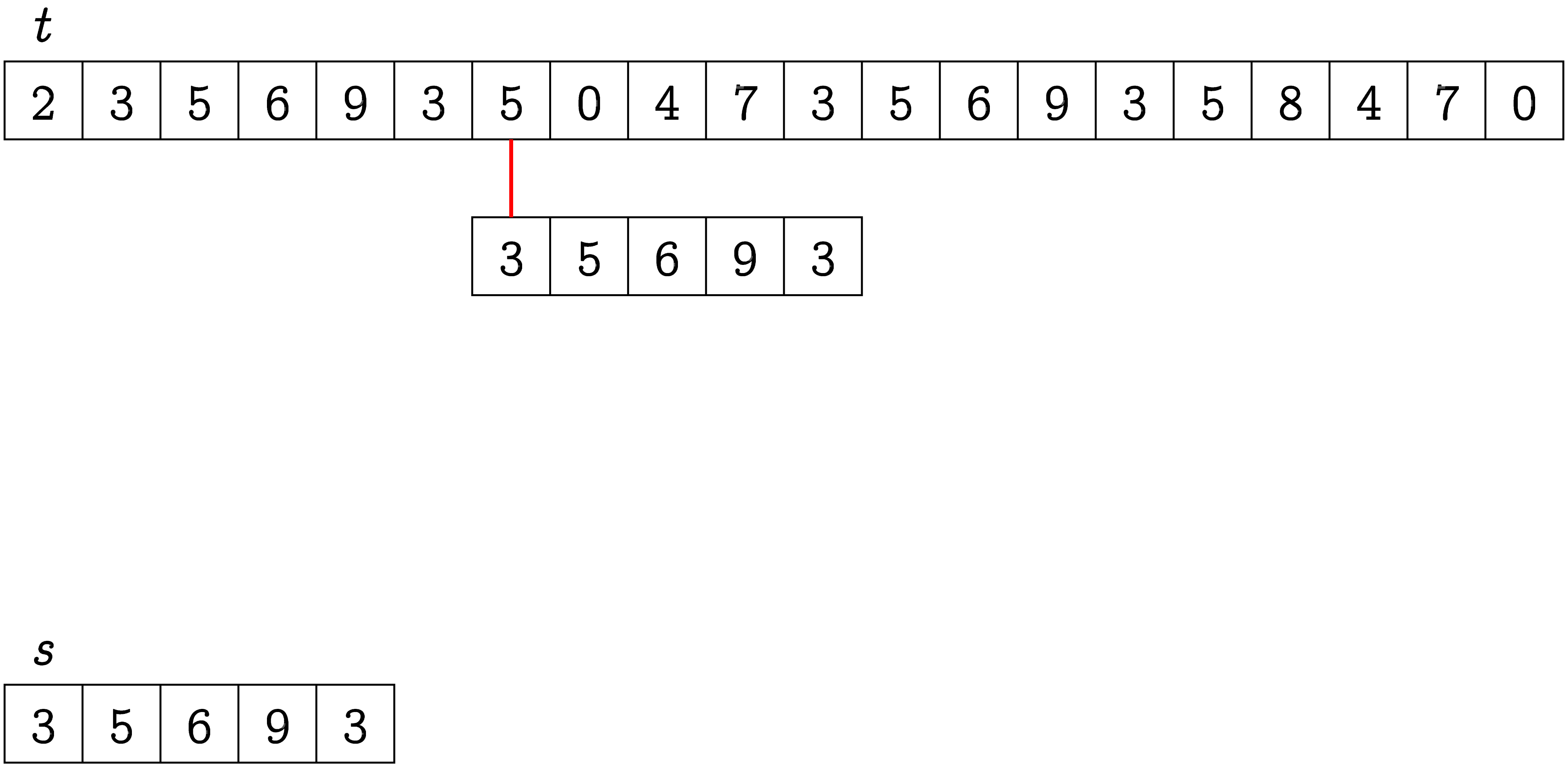
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 7$ $J = \{2\}$



t

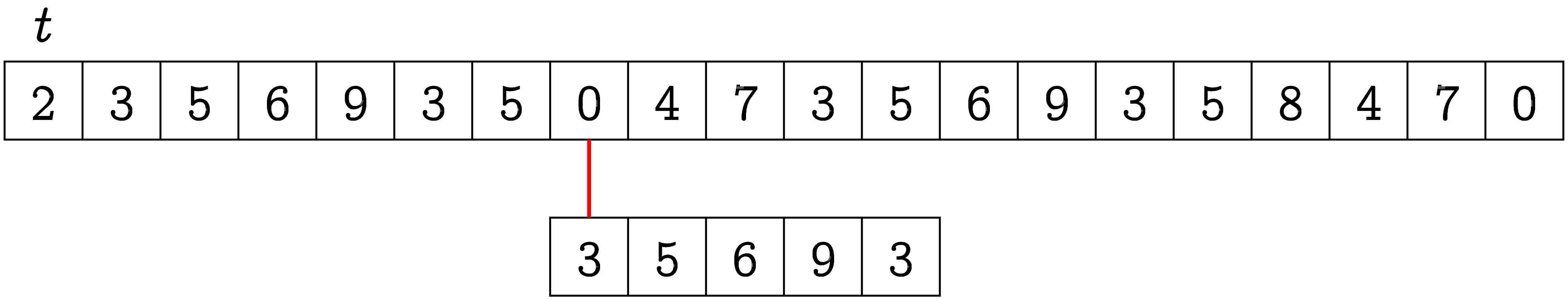
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 8$ $J = \{2\}$



t

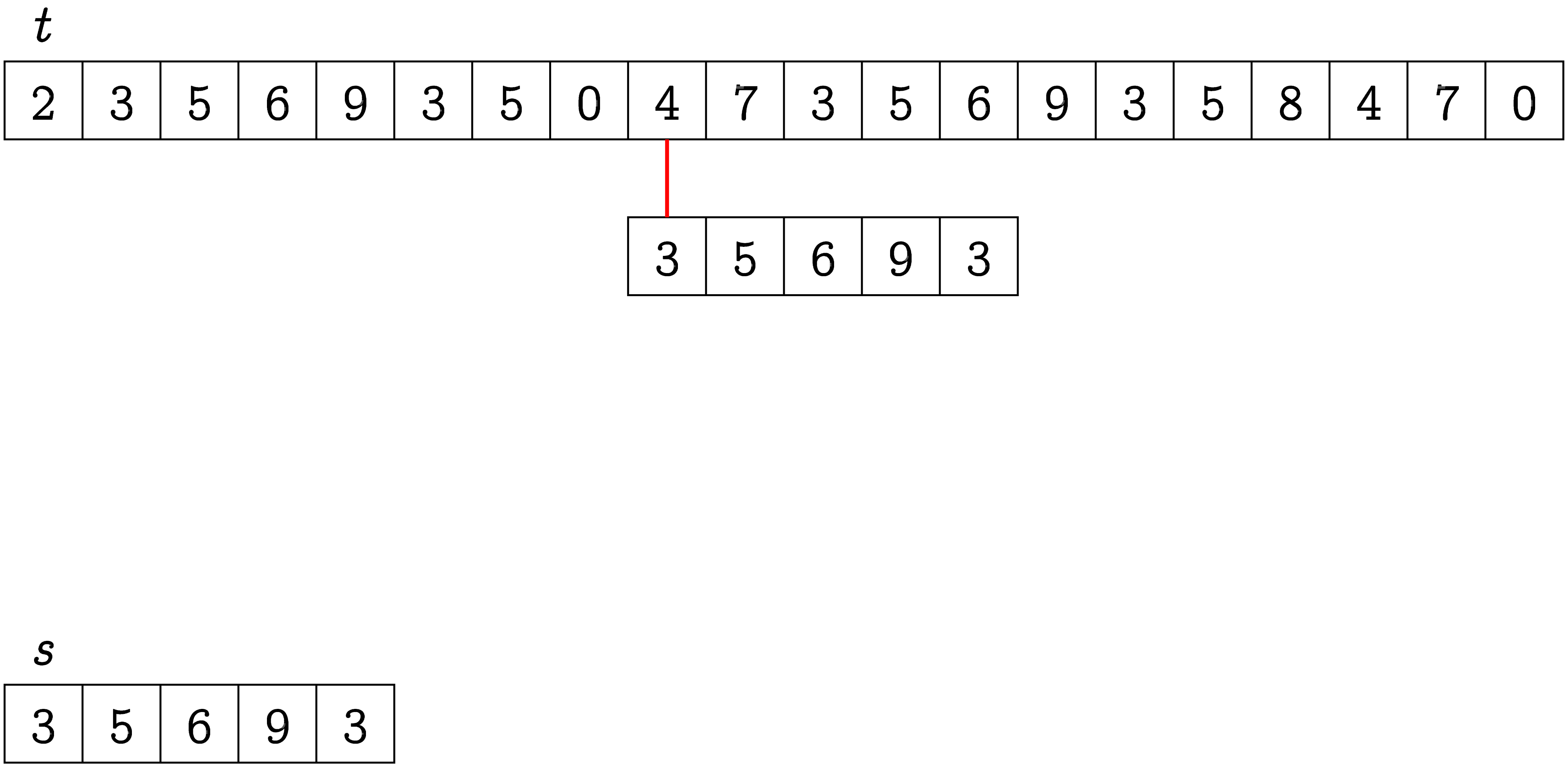
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 9$ $J = \{2\}$



t

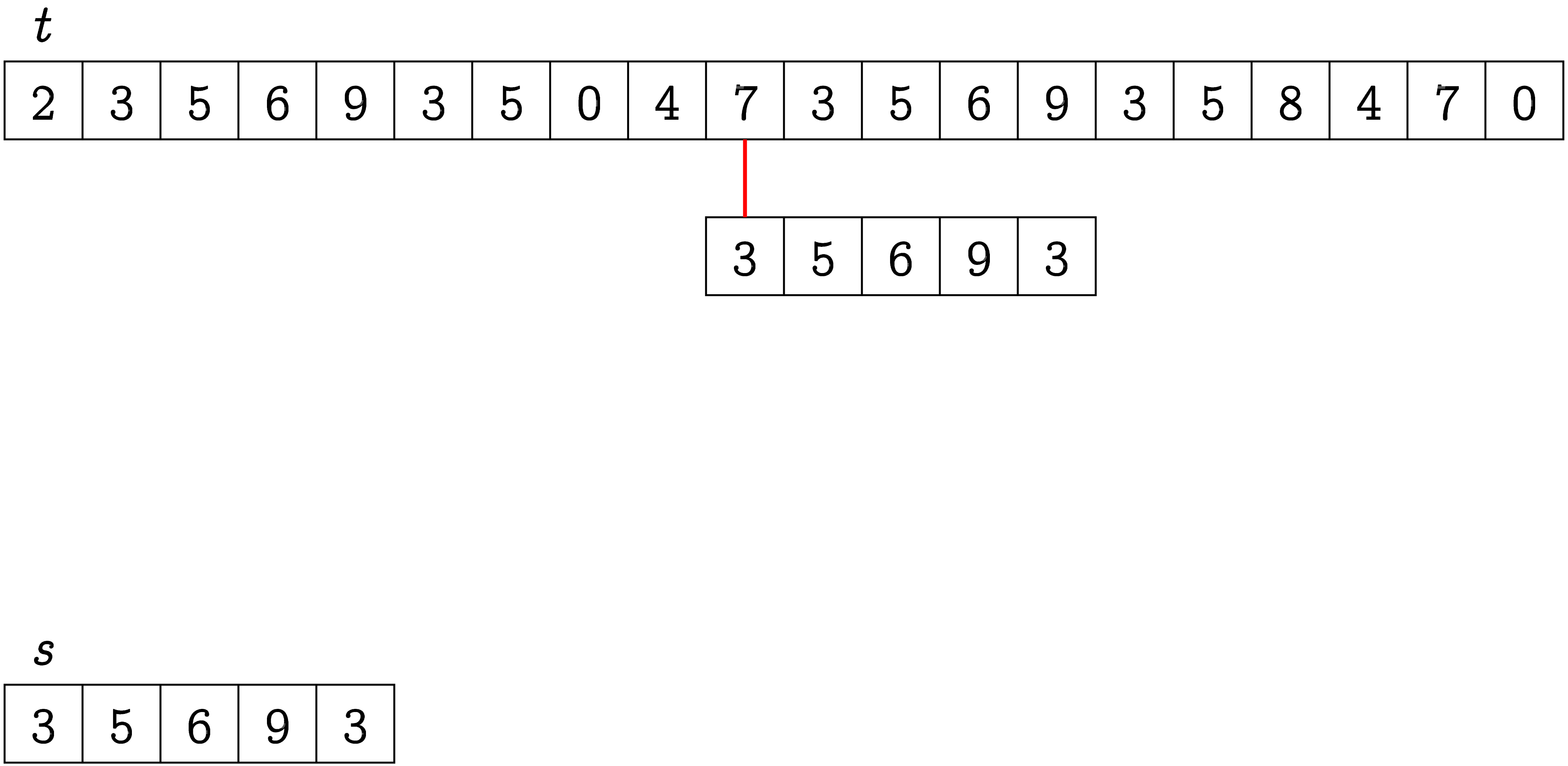
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 10$ $J = \{2\}$



t

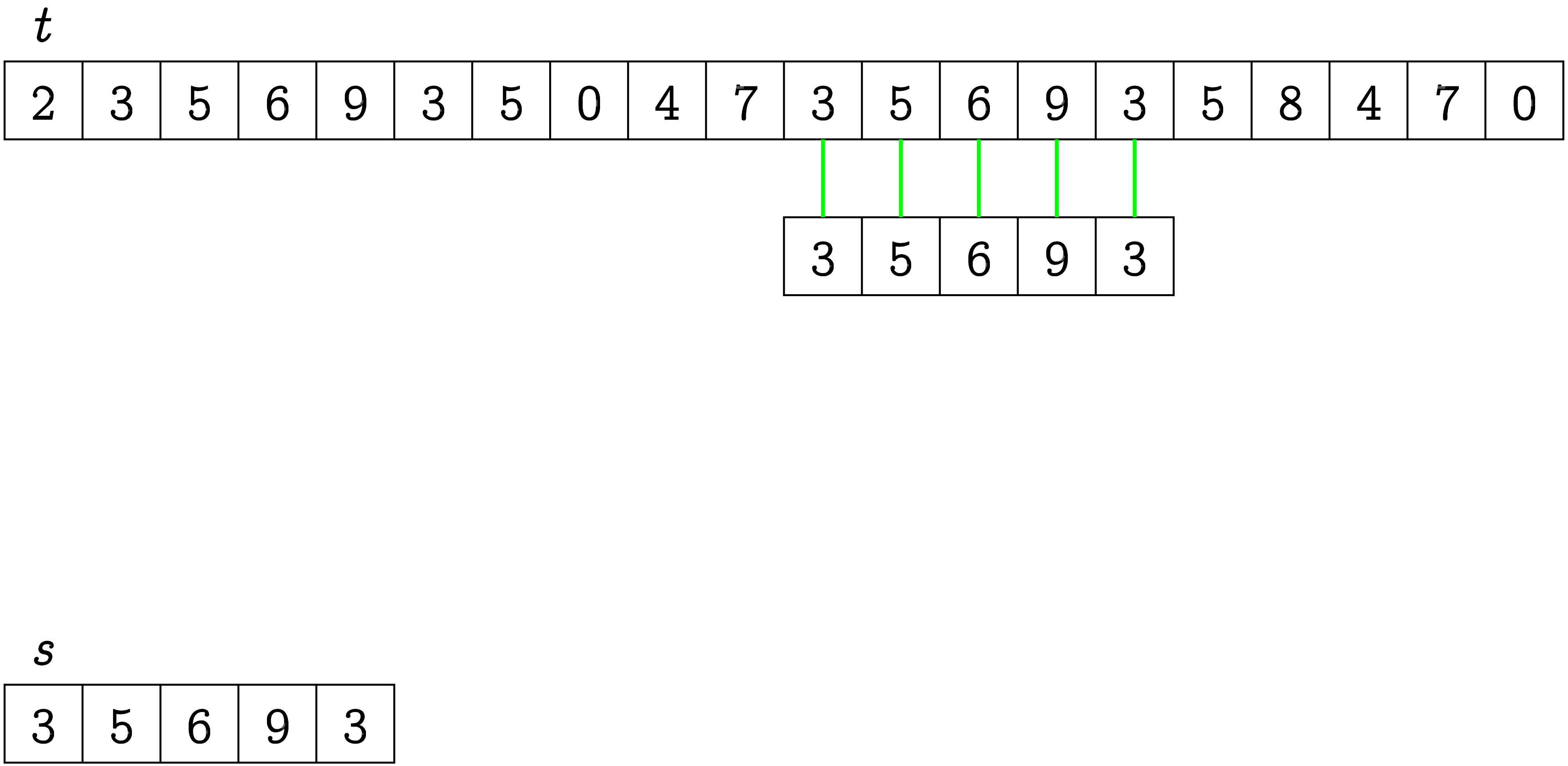
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

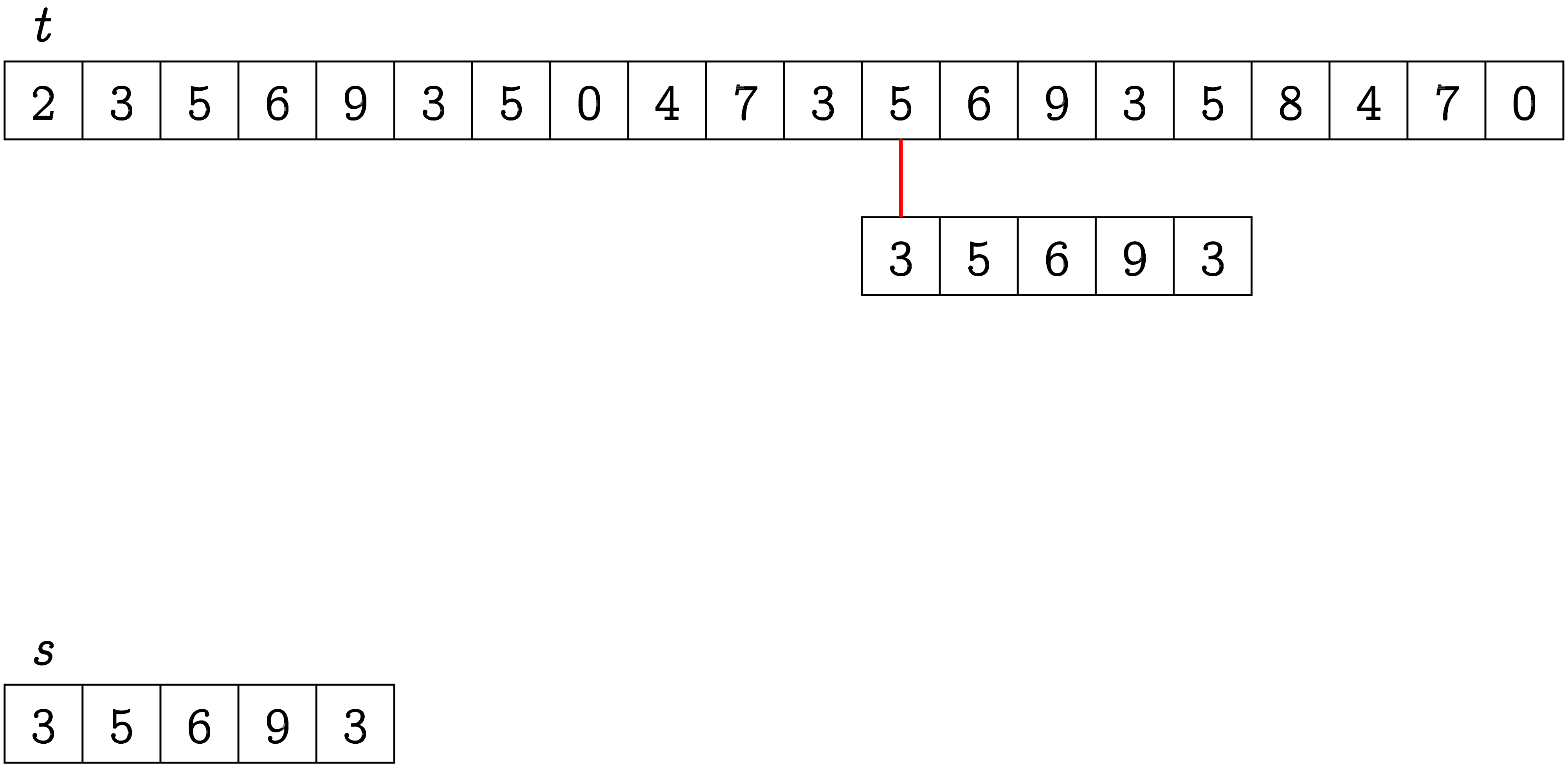
s

3	5	6	9	3

$i = 11$ $J = \{2, 11\}$



$i = 12$ $J = \{2, 11\}$



t

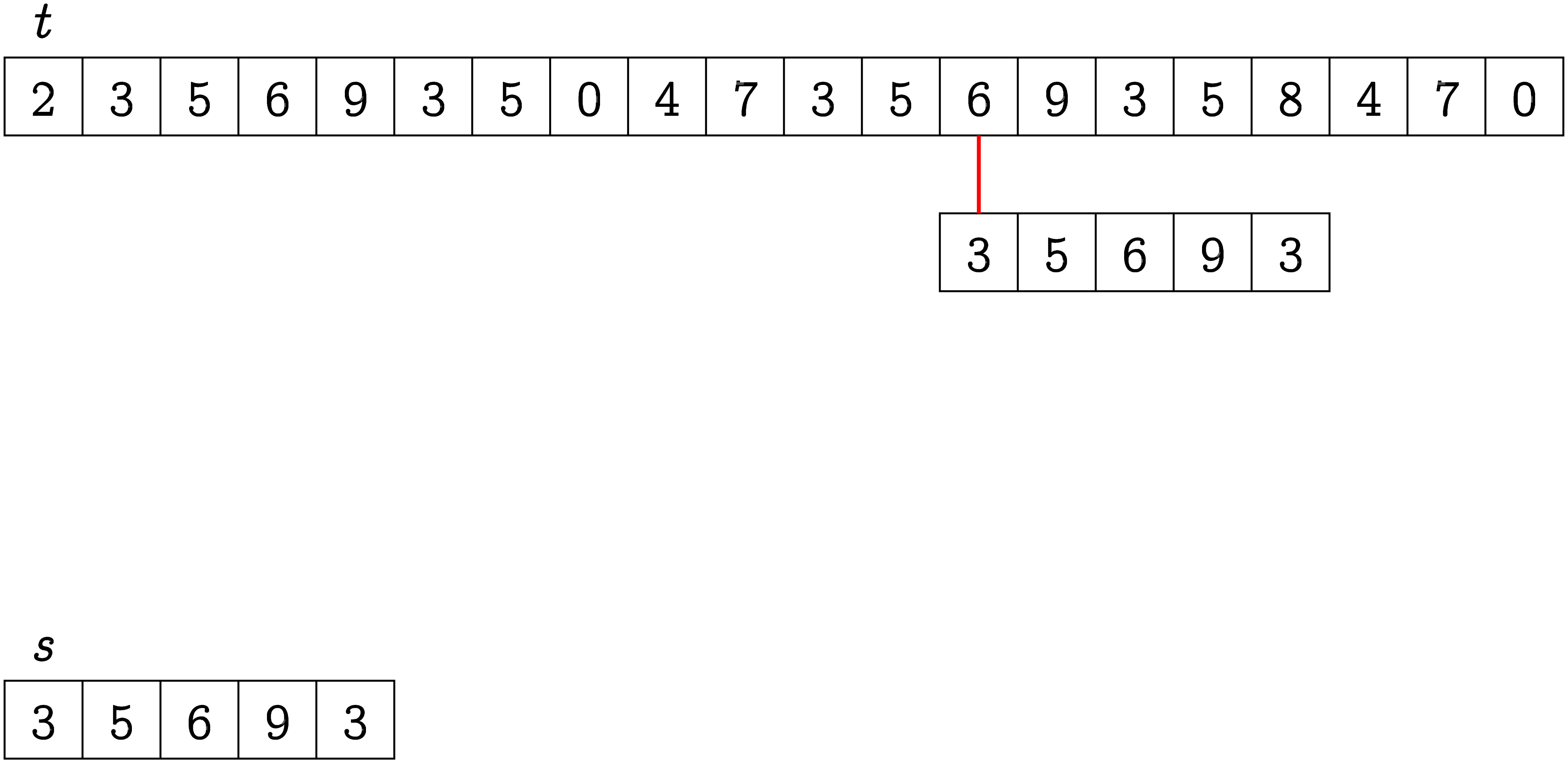
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 13$ $J = \{2, 11\}$



t

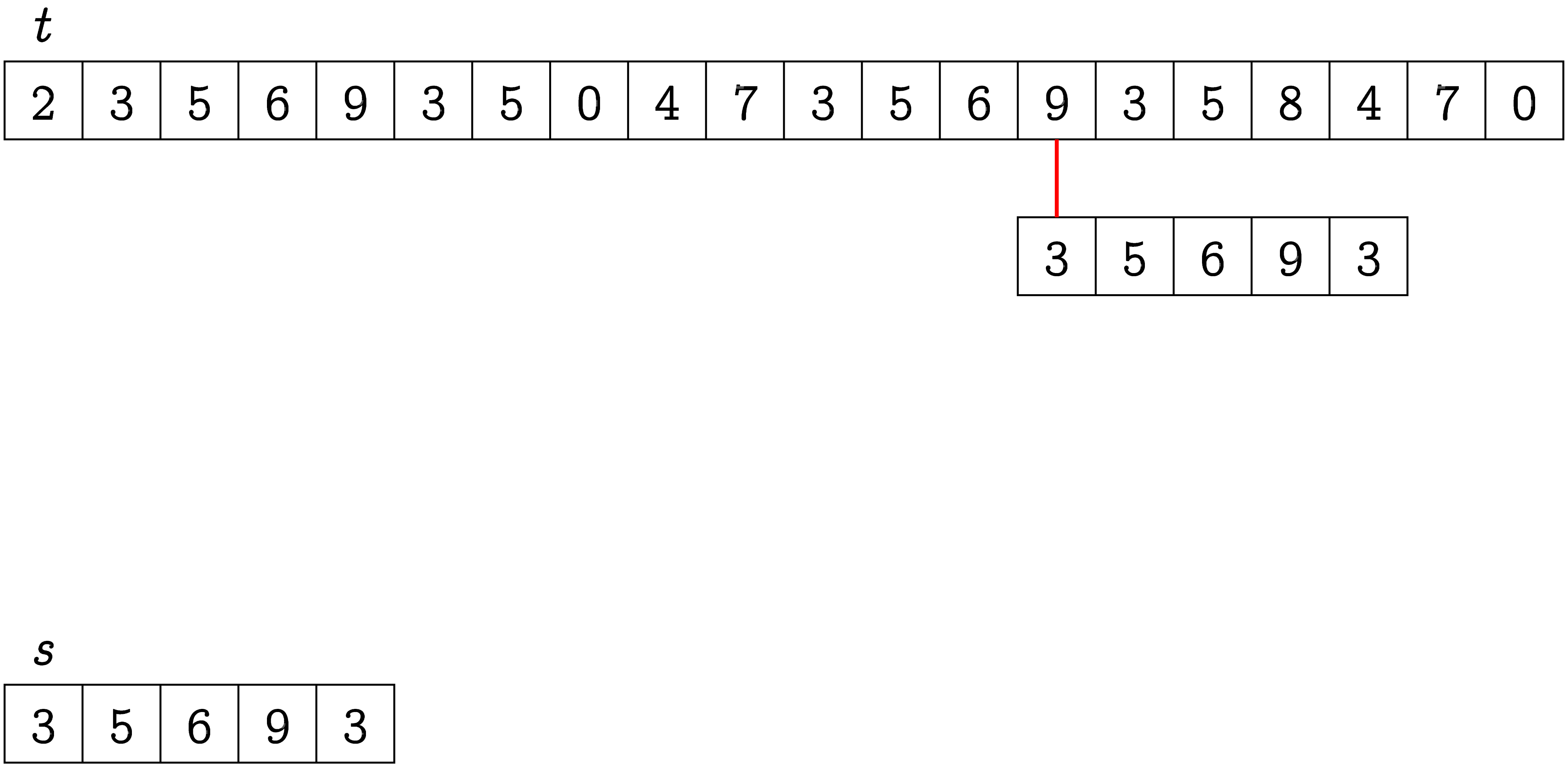
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$i = 14$ $J = \{2, 11\}$



t

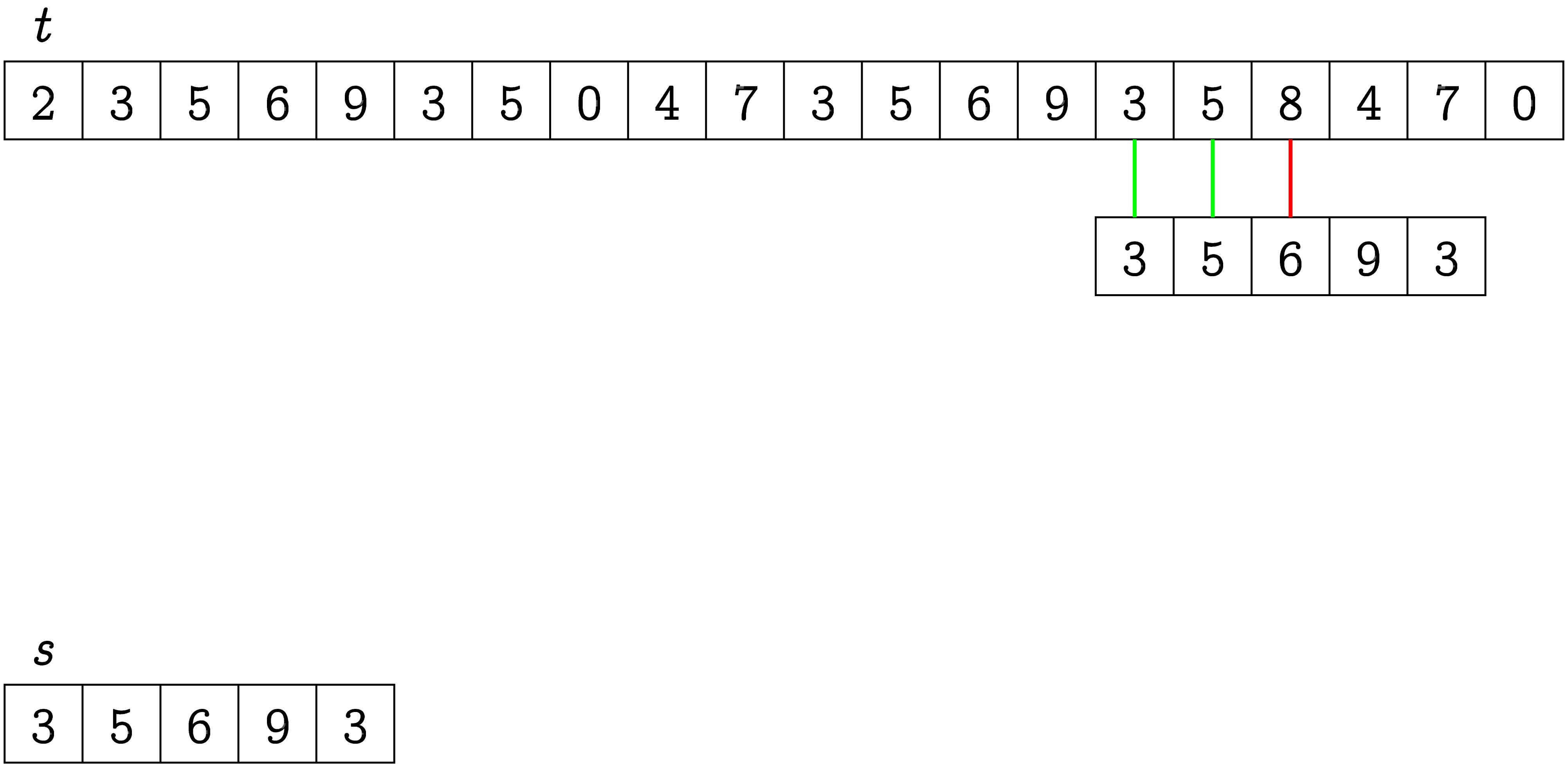
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

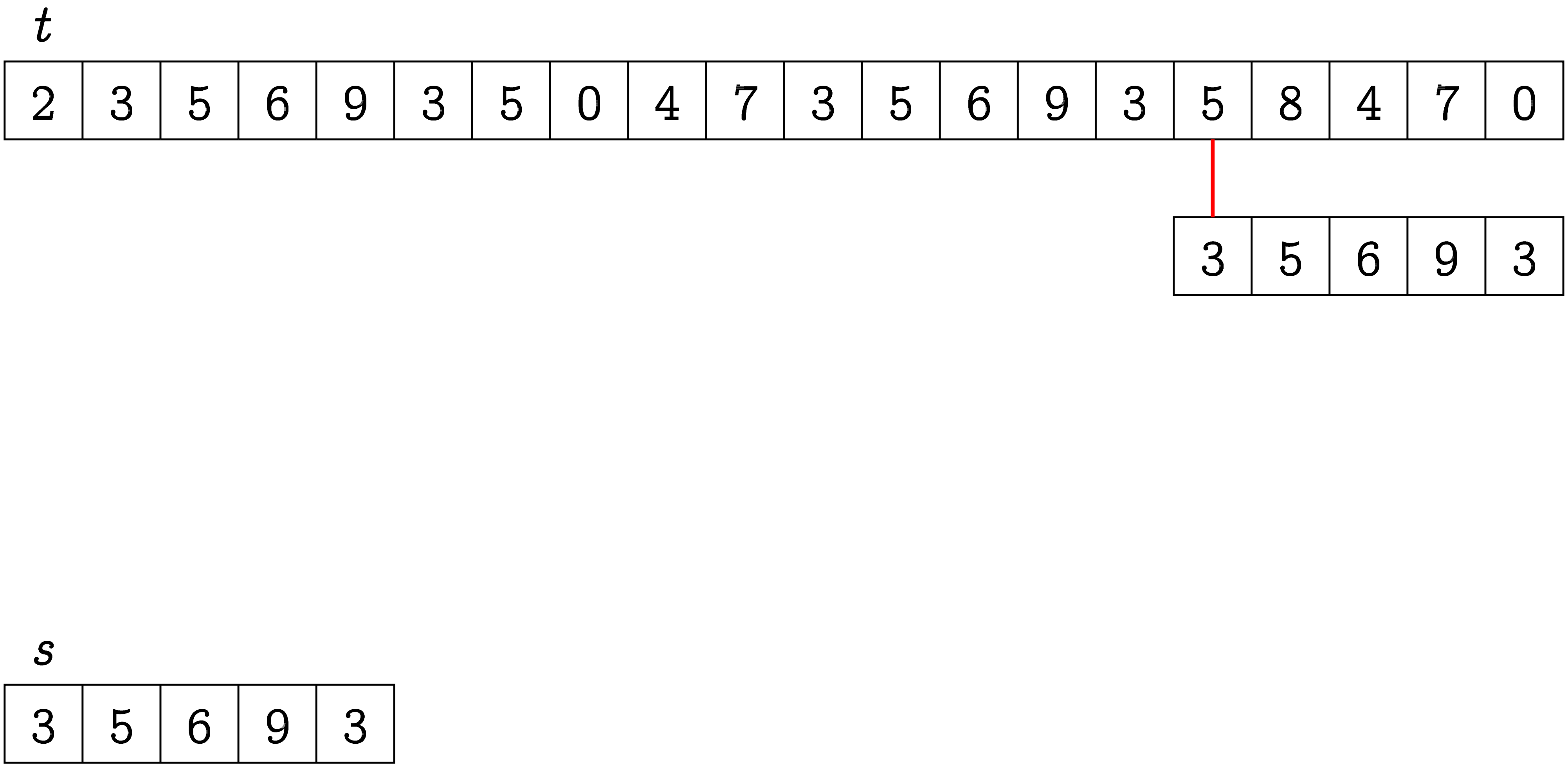
s

3	5	6	9	3

$i = 15$ $J = \{2, 11\}$



$i = 16$ $J = \{2, 11\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

Töö kiirendamiseks üritame kontrolle „$s = t[i \ldots i + m - 1]$?“ kiiremini teha.

Idee: defineerime mingi funktsiooni $h : \Sigma^m \longrightarrow X$ nii, et

- hulga X elementide võrdlemine on konstantse keerukusega;
- $h(t[i+1 \ldots i+m])$ on arvutatav $h(t[i \ldots i+m-1])$-st konstantses ajas.

Seejärel kontrollime iga i jaoks, kas $h(s) = h(t[i \ldots i + m - 1])$. Kui jah (mingi i jaoks), siis kontrollime, kas $s = t[i \ldots i + m - 1]$.

Asümptootiliselt me ei võida, kuid veendumine, et s t-s mingis positsioonis ei esine, käib üldiselt märksa kiiremini.

Kui $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, siis sõne pikkusega m on mingi (ülimalt) m-kohaline arv.

Üldjuhul (suvalise Σ korral) olgu $d = |\Sigma|$ ning olgu $\nu : \Sigma \longrightarrow \{0, \ldots, d-1\}$ mingi fikseeritud bijektsioon. Siis igale m-tähelisele sõnele vastab m-kohaline arv d-ndsüsteemis.

$$\nu(s) := \sum_{i=1}^{m} \nu(s[i]) d^{m-i}$$

Olgu $q \in \mathbb{N}$. Võtame $h(s) := \nu(s) \mod q$.

(q võtame võimalikult suure, kuid piisavalt väikse selleks, et $h(s)$ täisarvutüüpi ära mahuks.)

$h(t[i+1\dots i+m])$ on arvutatav $h(t[i\dots i+m-1])$-st konstantses ajas:

$$\begin{aligned} h(t[i+1\dots i+m]) = &\left(h(t[i\dots i+m-1]) - \nu(t[i]) \cdot d^{m-1}\right) \\ &\cdot d \\ &+ \nu(t[i+m]) \pmod q \enspace . \end{aligned}$$

Seejuures $d^{m-1} \mod q$ võib ette valmis arvutada.

Kõik arvutused on seejuures $\mod q$.

Vaatame eelmist näidet. Võtame $q = 17$.

Meil oli $s =$ „35693“. Siis $s \mod 17 = 10$.

Üldjuhul tuleb meil leida $\sum\limits_{i=1}^{m} \nu(s[i])d^{m-i} \mod q$. Seda on mugav teha Horneri skeemiga:

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 =$$

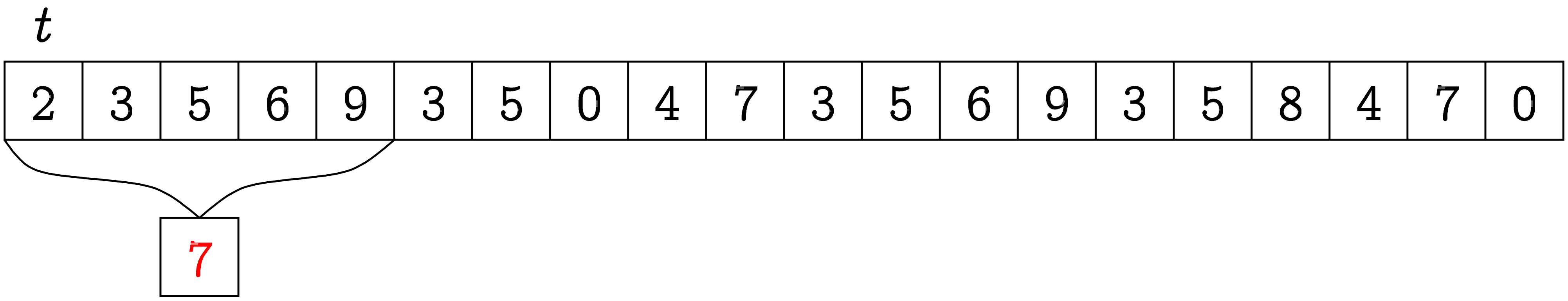
$$((\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_2)x + a_1)x + a_0\ .$$

(kõik arvutused ikka mod q)

Funktsioon modpoly(s, d, q, a, m) leiab sõnele $s[a \ldots a+m]$ vastava h väärtuse. Ta on:

```
$h := 0$
**for** $i := a$ **to** $a + m$ **do**
    $h := (((h \cdot d) \mod q) + \nu(s[i])) \mod q$
**return** $h$
```

$i = 1$ $J = \emptyset$



t

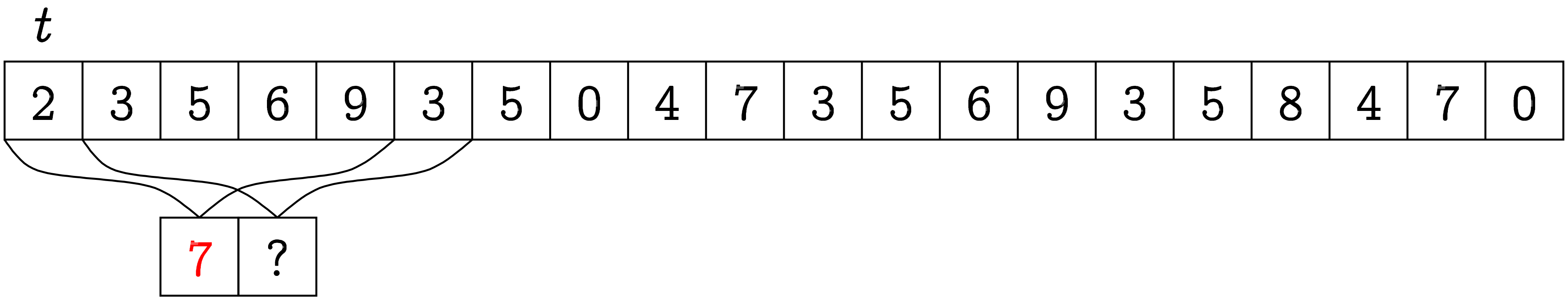
2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

s

3	5	6	9	3

mod 17 = 10

$i = 2$ $J = \emptyset$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

s

3	5	6	9	3

$\bmod 17 = 10$

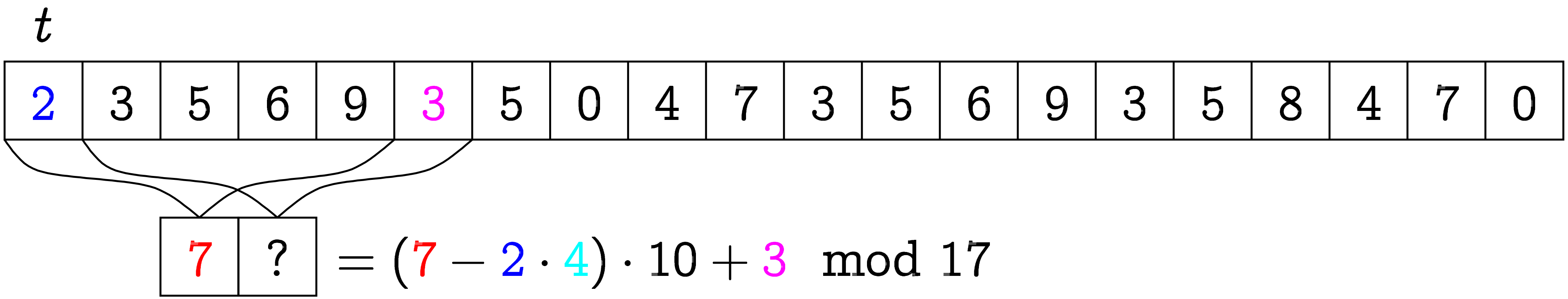
$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

$i = 2$ $J = \emptyset$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

7	?

$= (7 - 2 \cdot 4) \cdot 10 + 3 \mod 17$

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

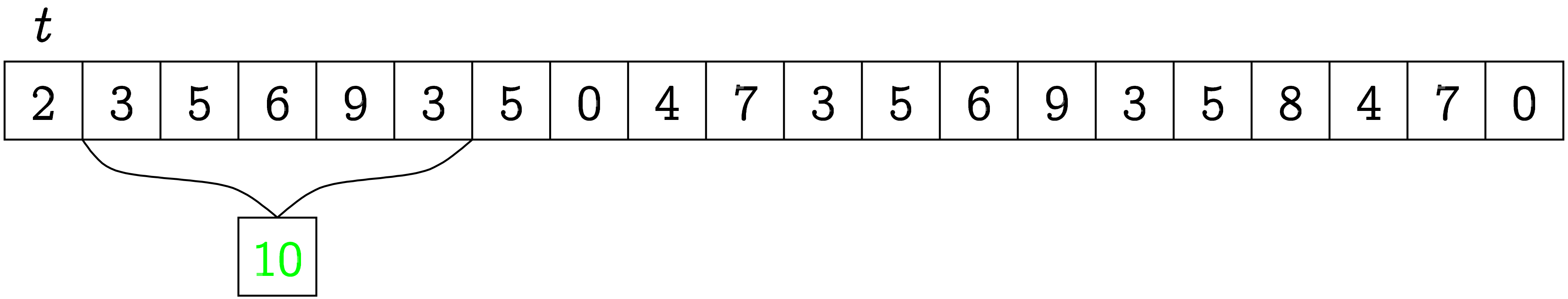
s

3	5	6	9	3

$\mod 17 = 10$

$i = 2$

$J = \emptyset$



s

3	5	6	9	3

$\mod 17 = 10$

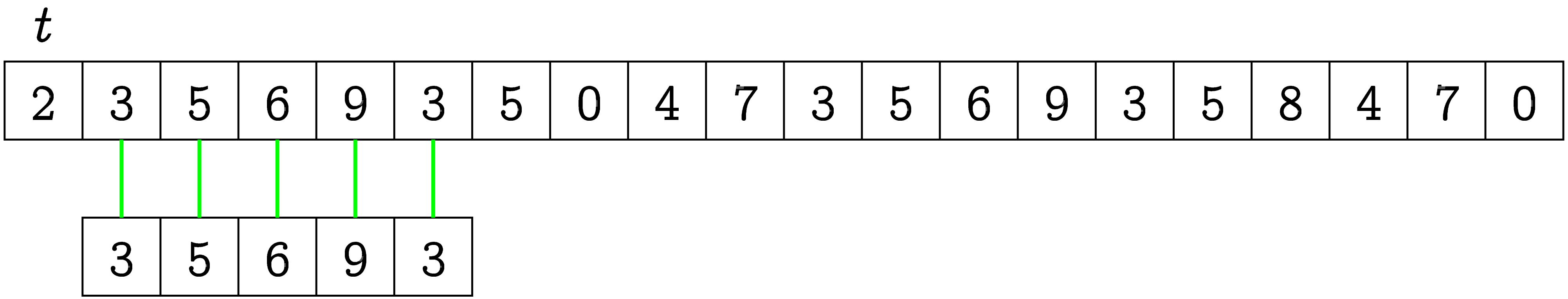
$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

$i = 2$ $J = \{2\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$\bmod 17 = 10$

$d = 10$

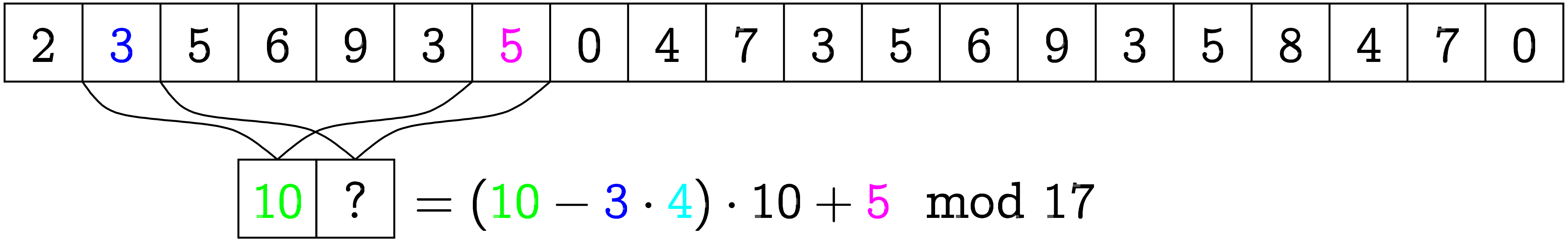
$m = 5$

$d^{m-1} = 10000$

$10000 \bmod 17 = 4$

$i = 3$ $J = \{2\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$$\boxed{10}\boxed{?} = (10 - 3 \cdot 4) \cdot 10 + 5 \mod 17$$

$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

3	5	6	9	3

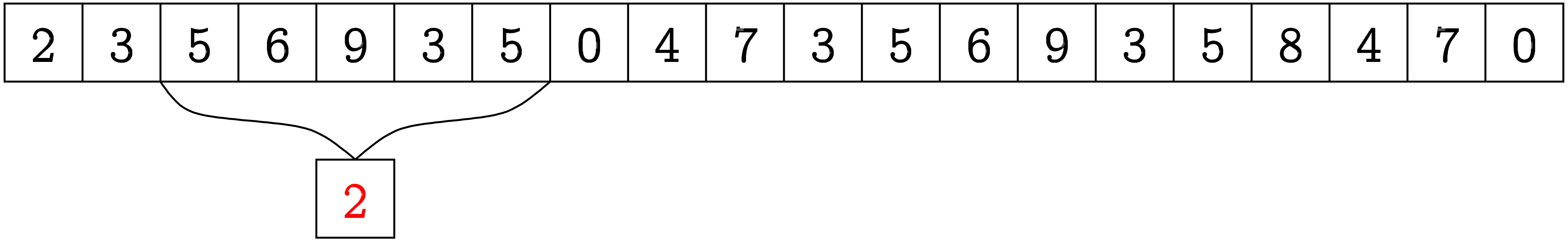
$\mod 17 = 10$

$10000 \mod 17 = 4$

$i = 3$

$J = \{2\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

3	5	6	9	3

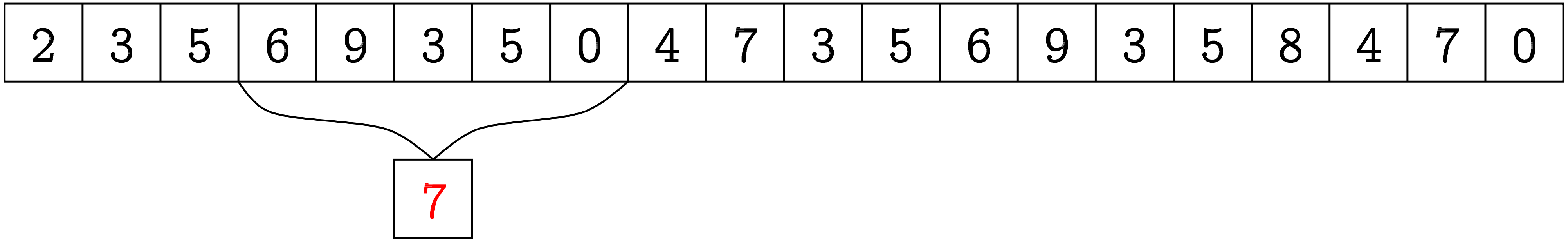
$\bmod 17 = 10$

$10000 \mod 17 = 4$

$i = 4$

$J = \{2\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

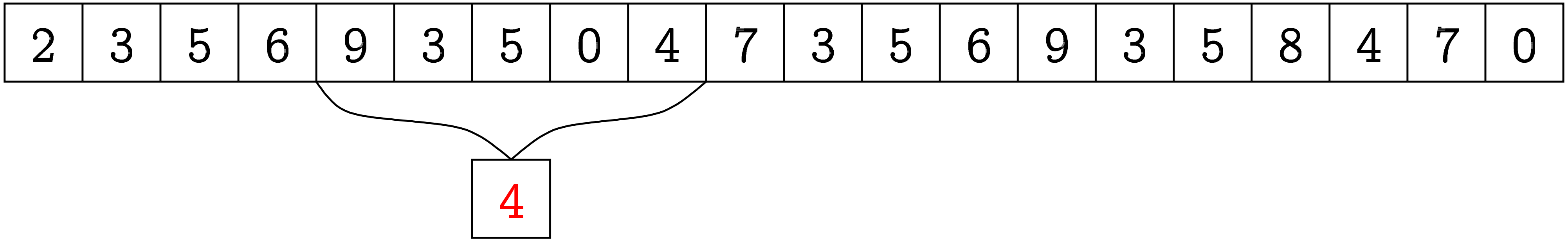
s

3	5	6	9	3

$\mod 17 = 10$

$i = 5$ $J = \{2\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

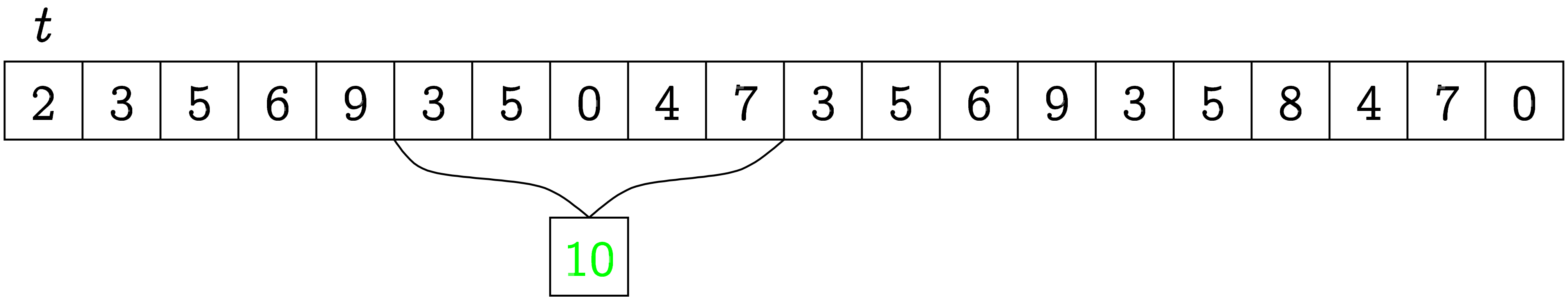
s

3	5	6	9	3

$\bmod 17 = 10$

$10000 \mod 17 = 4$

$i = 6$ $J = \{2\}$


t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

10

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

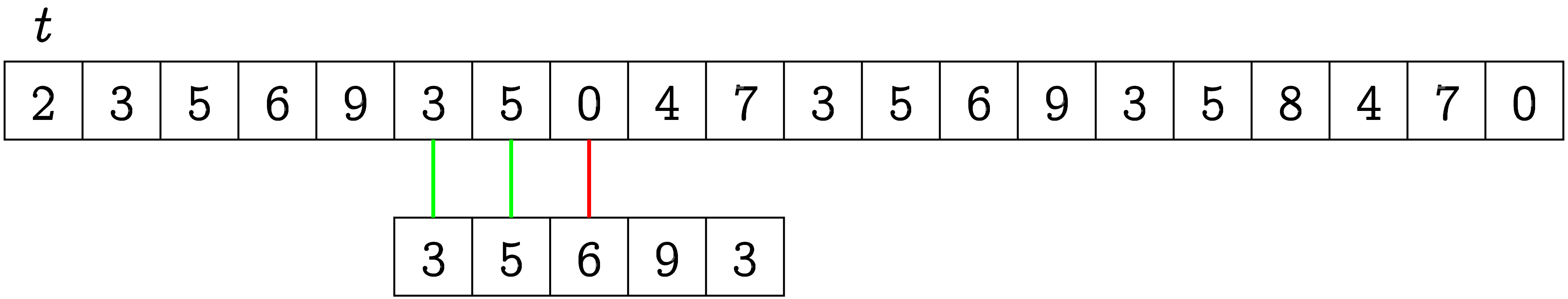
s

3	5	6	9	3

$\mod 17 = 10$

$i = 6$

$J = \{2\}$



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

s

3	5	6	9	3

$\bmod 17 = 10$

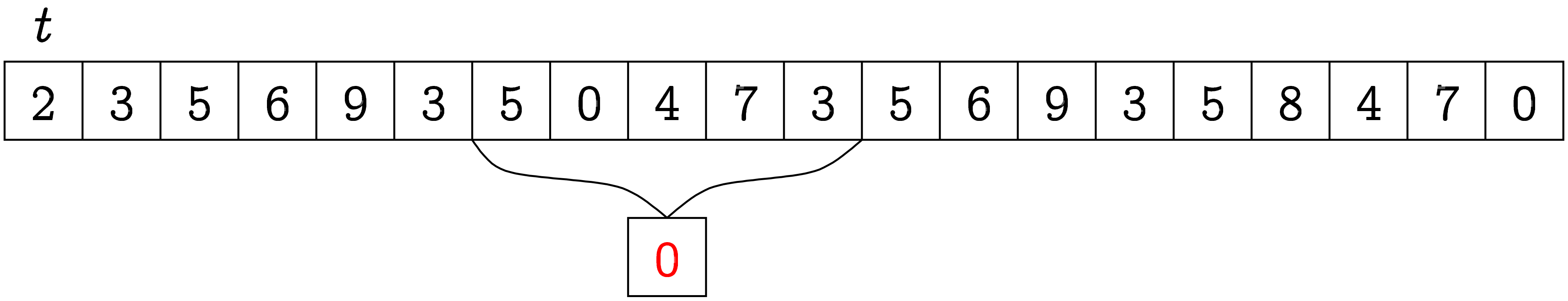
$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

$i = 7$ $J = \{2\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

s

3	5	6	9	3

$\bmod 17 = 10$

$d = 10$

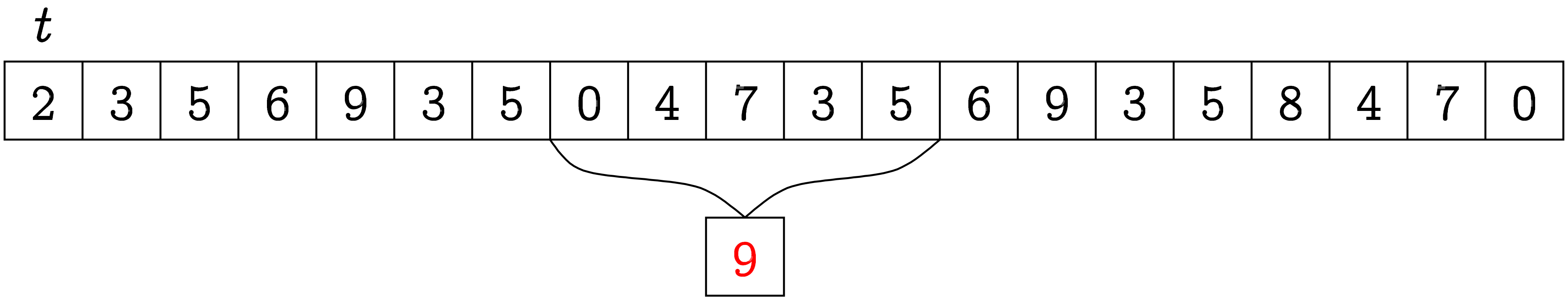
$m = 5$

$d^{m-1} = 10000$

$10000 \bmod 17 = 4$

$i = 8$

$J = \{2\}$



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

s

3	5	6	9	3

$\bmod 17 = 10$

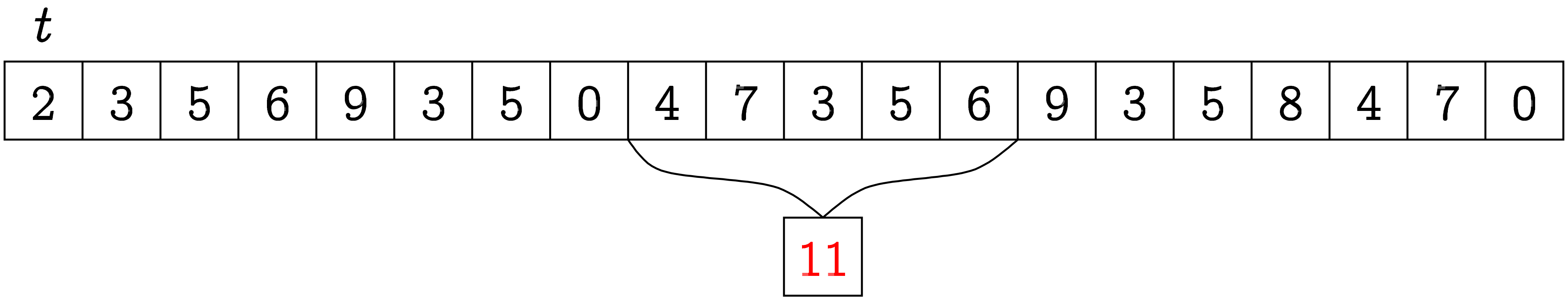
$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \bmod 17 = 4$

$i = 9$ $J = \{2\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

11

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

s

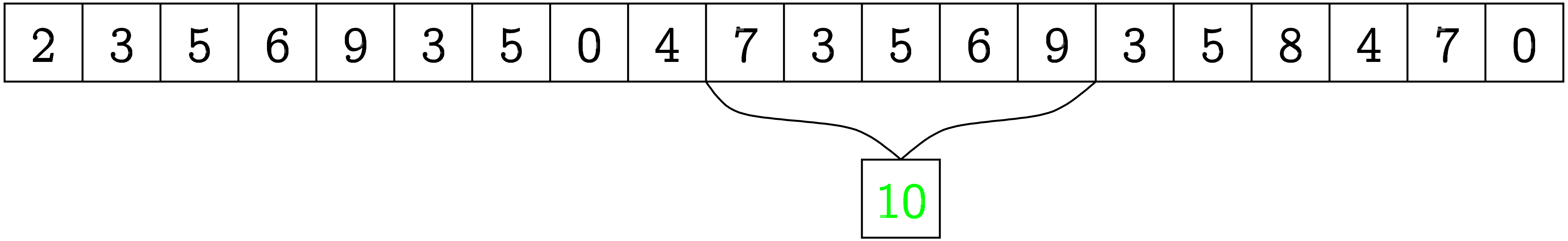
3	5	6	9	3

$\mod 17 = 10$

$i = 10$

$J = \{2\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

s

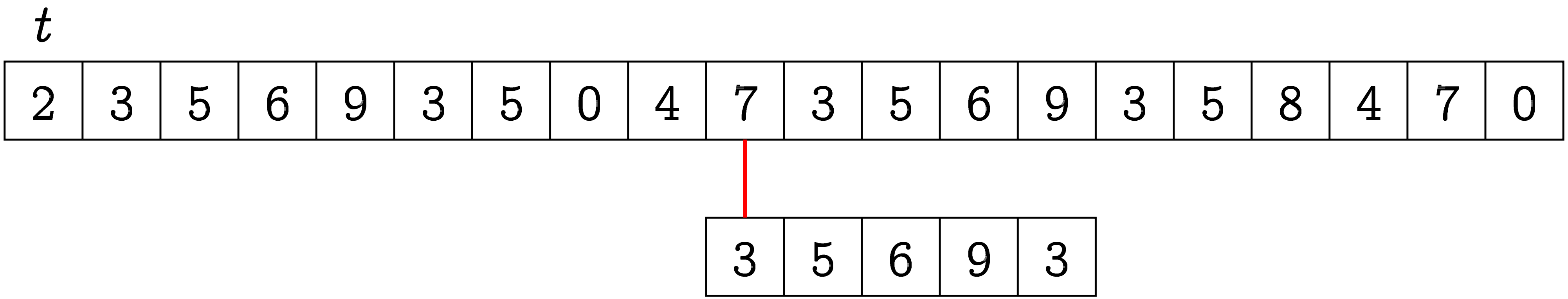
3	5	6	9	3

$\mod 17 = 10$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

$i = 10$ $J = \{2\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

$d = 10$

$m = 5$

$d^{m-1} = 10000$

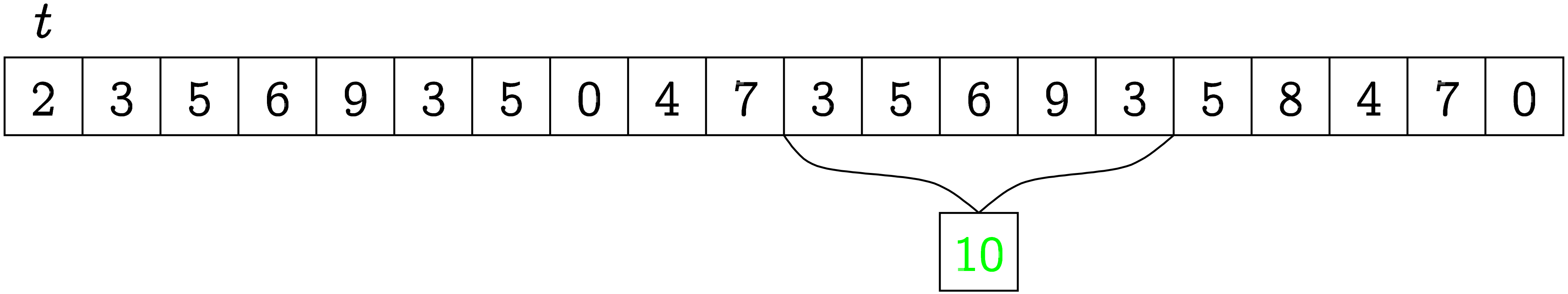
$10000 \mod 17 = 4$

s

3	5	6	9	3

$\mod 17 = 10$

$i = 11$ $J = \{2\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

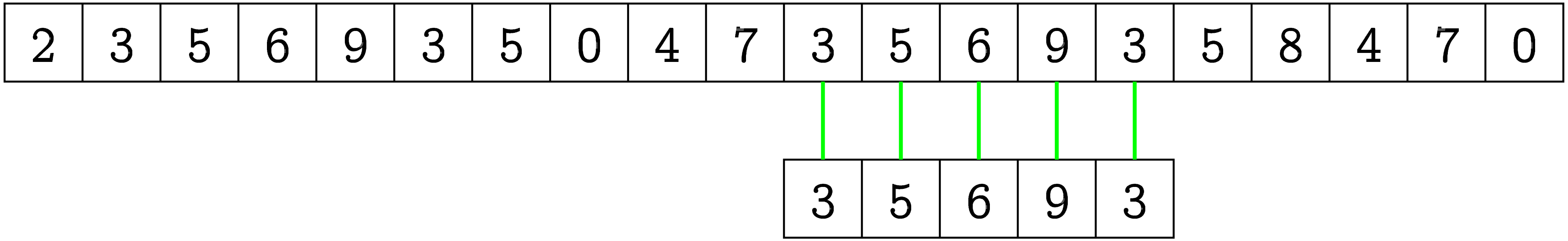
3	5	6	9	3

$\bmod 17 = 10$

$10000 \bmod 17 = 4$

$i = 11$ $J = \{2, 11\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

3	5	6	9	3

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

s

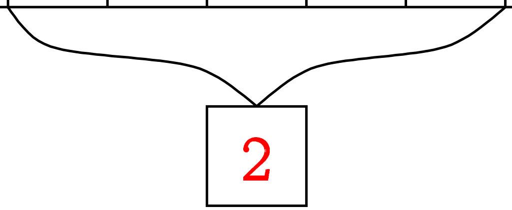
3	5	6	9	3

$\bmod 17 = 10$

$i = 12$ $J = \{2, 11\}$

t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0



$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

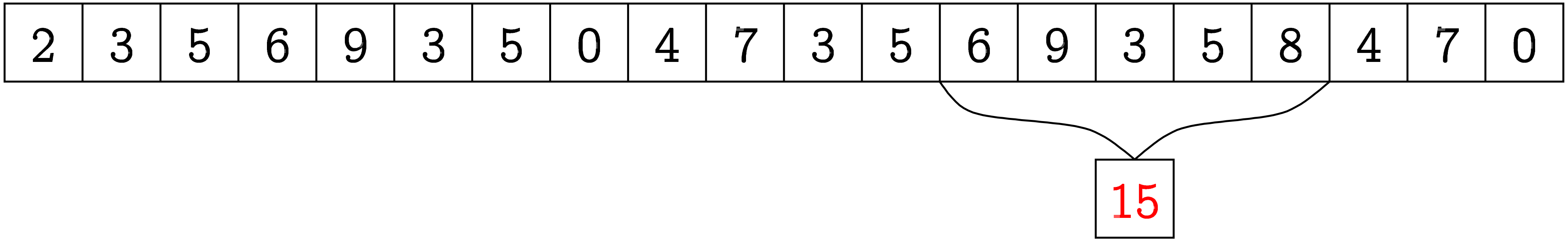
3	5	6	9	3

$\mod 17 = 10$

$10000 \mod 17 = 4$

$i = 13$ $J = \{2, 11\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

15

$d = 10$

$m = 5$

$d^{m-1} = 10000$

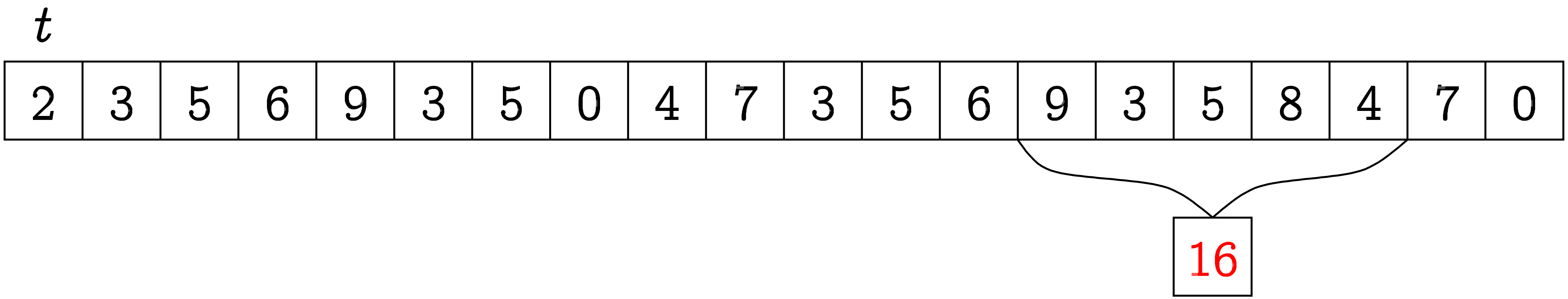
s

3	5	6	9	3

$\bmod 17 = 10$

$10000 \bmod 17 = 4$

$i = 14$ $J = \{2, 11\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

$10000 \mod 17 = 4$

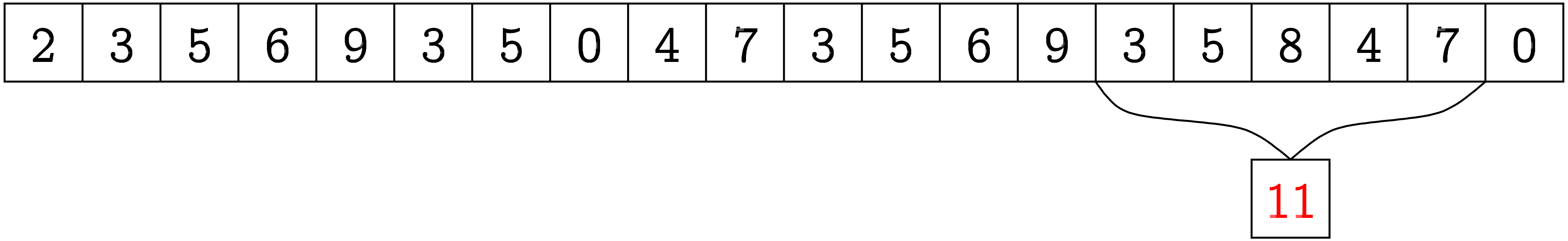
s

3	5	6	9	3

$\mod 17 = 10$

$i = 15$ $J = \{2, 11\}$

t



2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

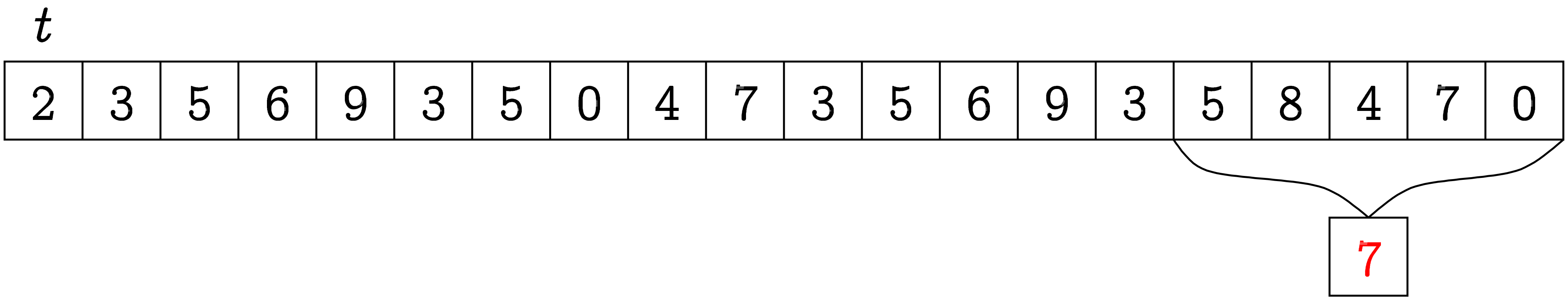
3	5	6	9	3

$\mod 17 = 10$

$10000 \mod 17 = 4$

$i = 16$

$J = \{2, 11\}$



t

2	3	5	6	9	3	5	0	4	7	3	5	6	9	3	5	8	4	7	0

$d = 10$

$m = 5$

$d^{m-1} = 10000$

s

3	5	6	9	3

$\bmod 17 = 10$

$10000 \mod 17 = 4$

Eelnev on tuntud kui **Rabin-Karpi** algoritm:

```
$J := \emptyset$; $m := |s|$; $n := |t|$; $d := |\Sigma|$
$D := 1$
for $i := 1$ to $m - 1$ do $D := D \cdot d \mod q$
$hs :=$ modpoly$(s, d, q, 1, m)$; $ht :=$ modpoly$(t, d, q, 1, m)$
for $i := 1$ to $n - m + 1$ do
    if $hs = ht$ then
        if võrdle_sõnesid$(s, m, t, i) = 0$ then $J \Leftarrow i$
    if $i < n - m + 1$ then
        $a1 := (ht - \nu(t[i]) \cdot D) \mod q$
        $a2 := a1 \cdot d \mod q$
        $ht := (a2 + \nu(t[i + m])) \mod q$
return $J$
```

Töö kiirendamiseks kasutame ära järgmist tähelepanekut:

Kui s ja $t[i \ldots i+m-1]$ erinevad esimest korda j-ndas positsioonis, siis $s[1] = t[i]$, $s[2] = t[i+1]$,…,$s[j-1] = t[i+j]$.

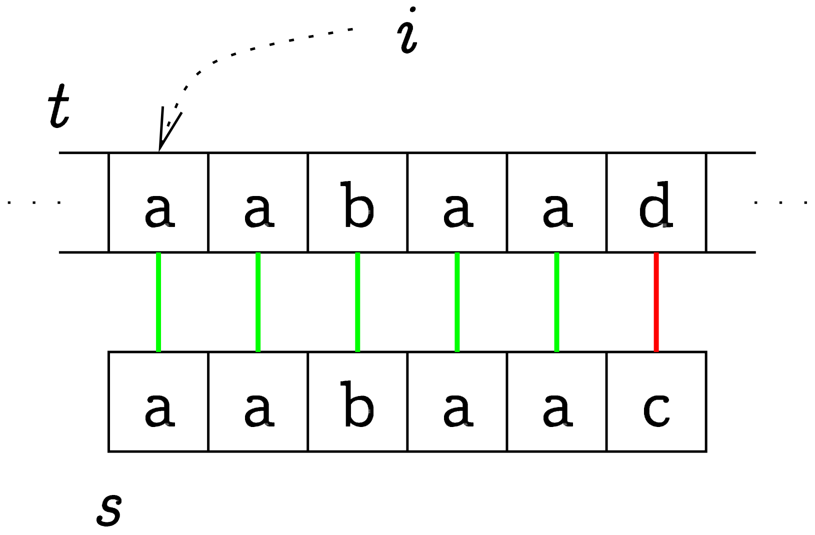
Sõltuvalt s-st võib järeldada, et mõningates järgmistes positsioonides ($i+1$, $i+2$, jne.) pole mõtet s-i t-s otsida — niikuinii ei ole.

Selleks, et s võiks t-s esineda positsioonis $i+1$, on tarvilik $s[1] = s[2]$, $s[2] = s[3]$,… $s[j-2] = s[j-1]$.

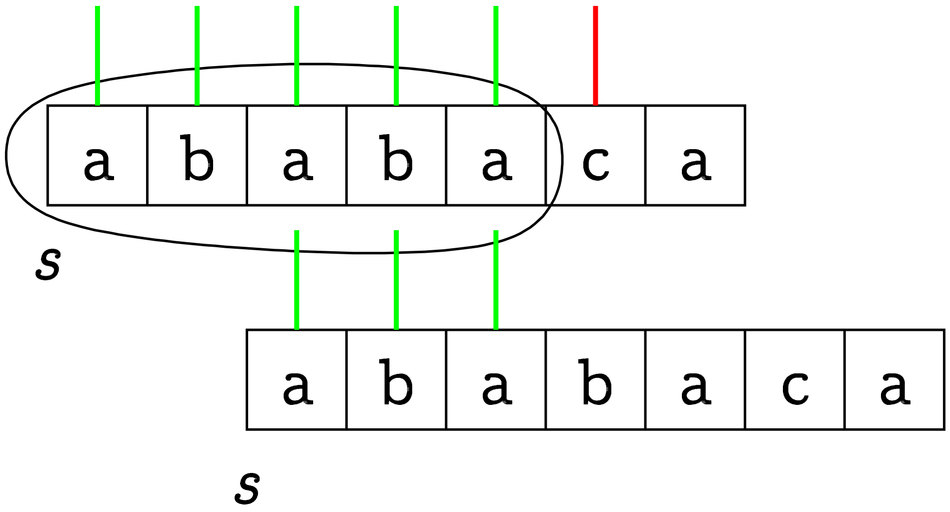
Selleks, et s võiks t-s esineda positsioonis $i+2$, on tarvilik $s[1] = s[3]$, $s[2] = s[4]$,… $s[j-3] = s[j-1]$.

Jne.

Näiteks:



Saades siin erinevuse 6. positsioonis, teame, et $i+1$ korral tuleb | |, $i+2$ korral tuleb |, $i+3$ korral tuleb | | ja edasi on mõtet võrrelda $s[3]$-e ja $t[(i+3)+2]$-e.



Kui erinevus tekkis j-ndas positsioonis, siis vaatame sõnet $s[1 \ldots j-1]$.

Nihutame sõnet s paremale senikaua, kuni tema algus langeb kokku sõne $s[1 \ldots j-1]$ lõpuga.

S.t. me otsime pikimat sellist sõne u, mis oleks $s[1 \ldots j-1]$ prefiksiks ja sufiksiks.

Olgu $\pi_1, \ldots, \pi_m$ massiiv („prefiksfunktsioon“), nii et iga i jaoks on

- $\pi_i \in \{0, 1, \ldots, i-1\}$;
- $s[1 \ldots \pi_i] \sqsupset s[1 \ldots i]$;
- π_i on suurim arv, mis eelmisi tingimusi rahuldab.

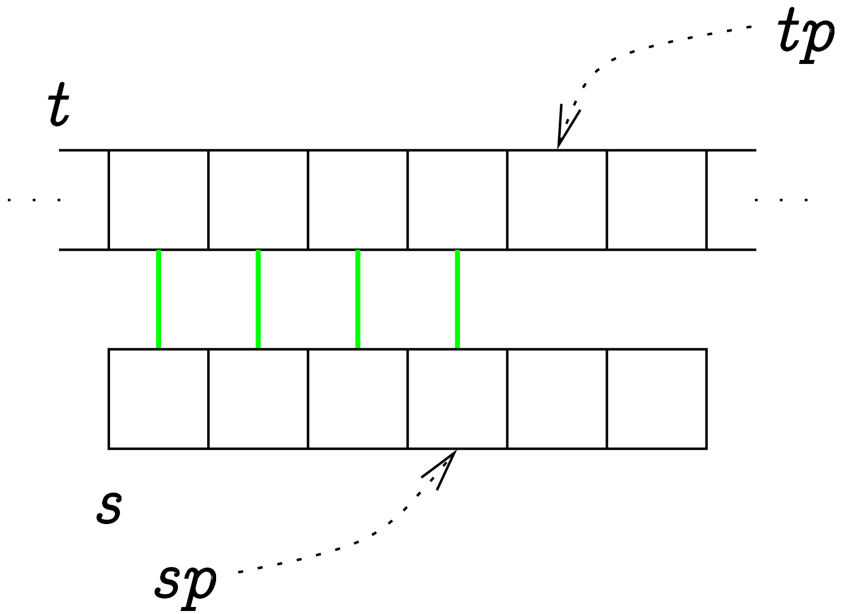
Näiteks kui $s =$ „aabcaabca“, siis π on:

i	1	2	3	4	5	6	7	8	9
π_i	0	1	0	0	1	2	3	4	5

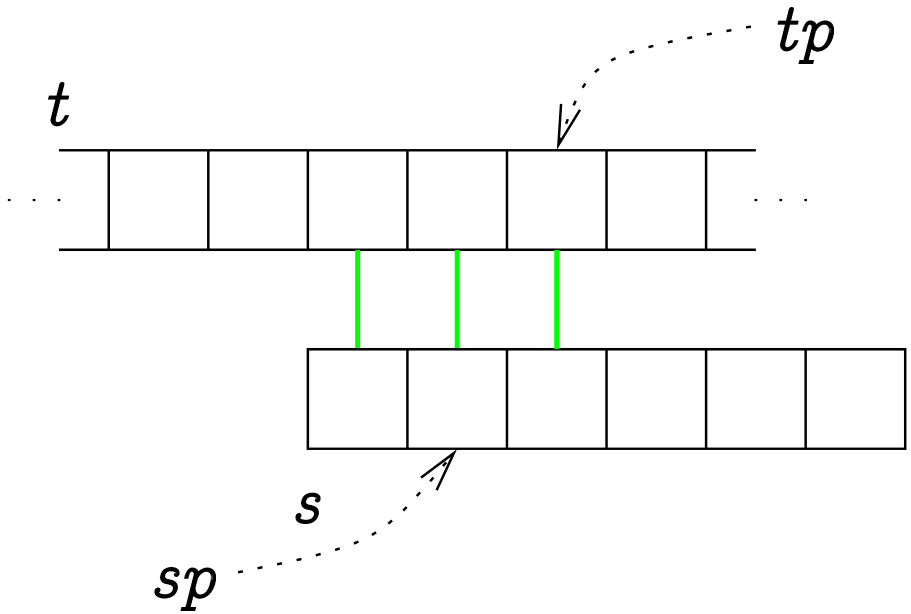
Alamsõne esinemisi saab siis otsida järgnevalt: (**Knuth-Morris-Pratti** (KMP) algoritm)

```
$m := |s|;\ n := |t|;\ J := \emptyset$
$\pi :=$ leia_pi($s$)
$sp := 0$
for $tp := 1$ to $n$ do
    while $sp > 0$ and $s[sp + 1] \neq t[tp]$ do $sp := \pi_{sp}$
    if $s[sp + 1] = t[tp]$ then $sp := sp + 1$
    if $sp = m$ then
        $J \Leftarrow tp - m + 1$
        $sp := \pi_{sp}$
return $J$
```

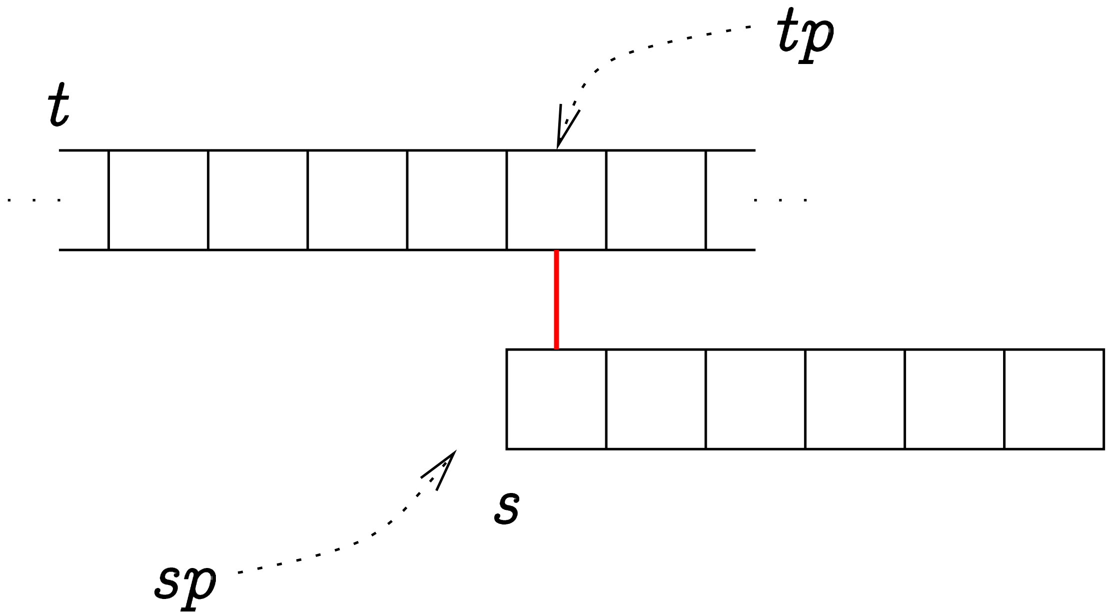
Peale 4. rida:



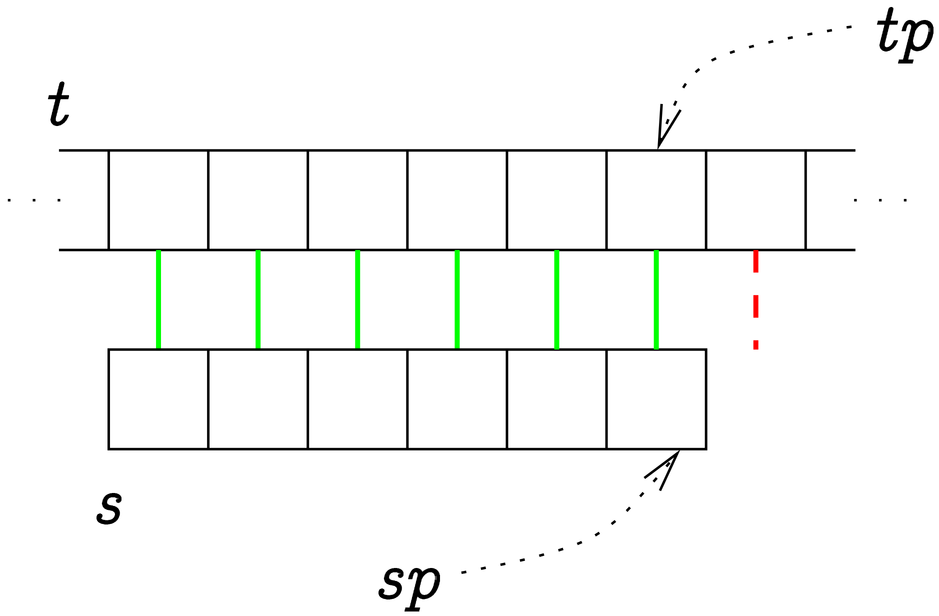
Peale 5. rida: kas



với



Read 7–9: kui $sp = m$, siis



Siis peaks $s[sp+1]$ võrdlemine t elemendiga negatiivse vastuse andma. Kuna $s[sp+1]$ tegelikult ei eksisteeri, siis tuleb seda juhtu eraldi töödelda (mitte teha järgmisel iteratsioonil reas 5).

Keerukus: ilma leia_pi väljakutseta on KMP-algoritmi tööaeg $O(n)$.

Näitame, et põhitsükli ühe iteratsiooni (read 4–9) *amortiseeritud* tööaeg on $O(1)$.

Potentsiaaliks võtame muutuja sp väärtuse. Küllalt suure ajaühiku korral siis:

Read 4 ja 7–9 võtavad $\leqslant 1$ ühiku tõelist aega. Kuna sp väärtus neis ridades ei suurene, siis võtavad nad ka $\leqslant 1$ ühiku amortiseeritud aega.

```
4   for $tp := 1$ to $n$ do
    ...
7       if $sp = m$ then
8           $J \Leftarrow tp - m + 1$
9           $sp := \pi_{sp}$
```

Rida 6 võtab $\leqslant 1$ ühiku tõelist aega. Kuna *sp* võib suureneda 1 võrra, siis võtab ta $\leqslant 2$ ühikut amortiseeritud aega.

```
6    if $s[sp+1] = t[tp]$ then $sp := sp + 1$
```

5. reas võtab iga iteratsioon $\leqslant 1$ ühiku tõelist aega. Et aga $\pi_i < i$, siis väheneb sp igal iteratsioonil (vähemalt ühe võrra). Seega võtab 5. rea iga iteratsioon $\leqslant 0$ ühikut amortiseeritud aega.

```
5      while $sp > 0$ and $s[sp+1] \neq t[tp]$ do $sp := \pi_{sp}$
```

Kokku võtab üks iteratsioon seega $\leqslant 1 + 2 + 0 = 3$ ühikut amortiseeritud aega.

Töö alguses $sp = 0$ ja alati $sp \geqslant 0$. Seega on tõeline koguaeg mitte suurem kui amortiseeritud koguaeg.

Korrektsuse näitamiseks tuleb näidata, et

1. s esineb t-s kõigil J-i lisatavatel positsioonidel;
2. ükski selline positsioon ei jää J-i lisamata.

Esimene neist järeldub otseselt invariandist peale 4. rida.

Teine järeldub sellest, et kui

$$s[1] = t[i], s[2] = t[i+1], \ldots, s[j] = t[i+j-1],$$

siis iga täiendava nihke k jaoks, mis on väiksem kui $j - \pi_j$, leidub $l \in \{1, \ldots, j-k\}$ nii, et $s[l] \neq s[l+k]$, s.t. $s[l] \neq t[i+k+l-1]$, s.t. s ei esine t-s positsioonil $i+k$.

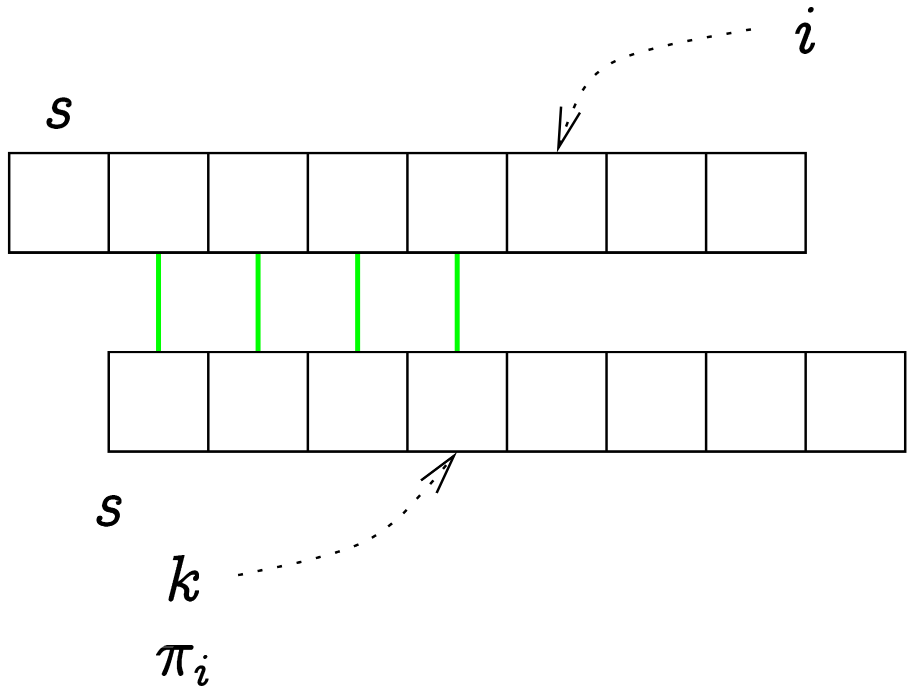
leia_pi(s) on sarnane otsimisalgoritmi endaga:

```
$m := |s|$; $\pi_1 := 0$
$k := 0$
for $i := 2$ to $m$ do
    while $k > 0$ and $s[k+1] \neq s[i]$ do $k := \pi_k$
    if $s[k+1] = s[i]$ then $k := k+1$
    $\pi_i := k$
return $\pi$
```

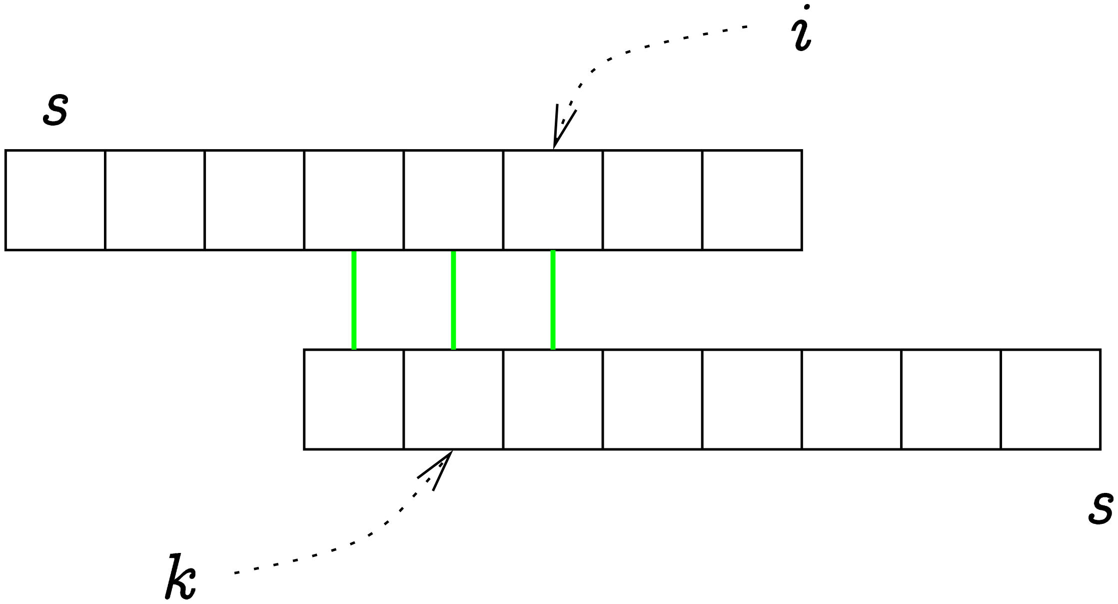
Taas saame näidata, et tsükli üks iteratsioon võtab $O(1)$ ühikut amortiseeritud aega. Potentsiaaliks on k väärtus.

Kogu KMP-algoritmi keerukus on seega $O(n+m)$.

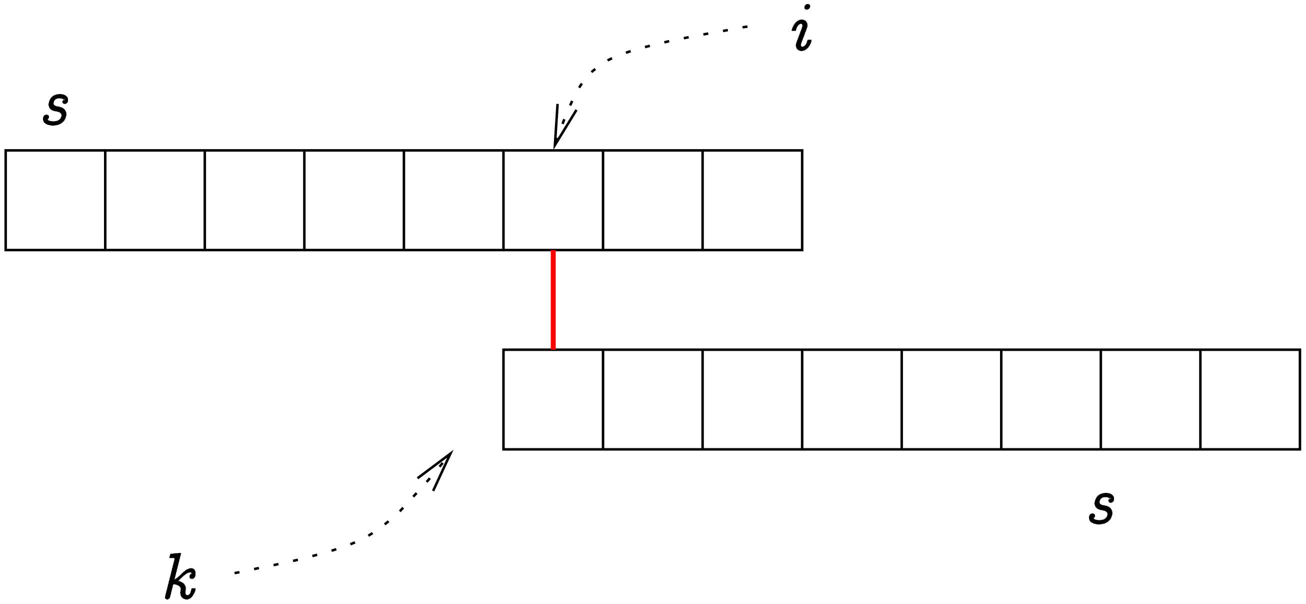
Peale 3. rida:



Peale 4. rida: kas



với



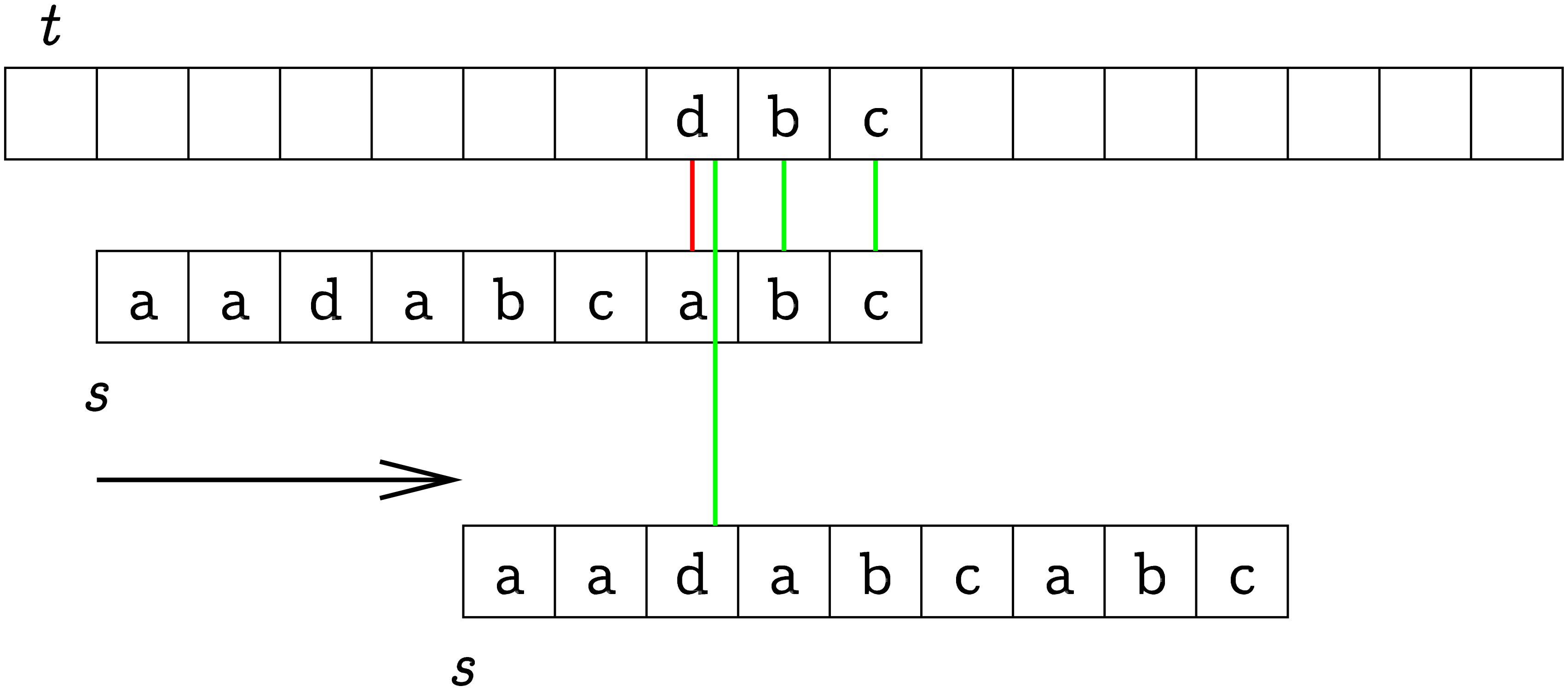
Boyer-Moore'i (BM) alamsõne otsimise algoritm töötab halvimal juhul ajas $O(m(n-m))$, aga pika s-i ja suure Σ korral võib praktikas kõige kiiremaks algoritmiks osutuda.

```
$m := |s|$; $n := |t|$; $J := \emptyset$
initsialiseeri_BM
$i := 1$
while $i \leqslant n - m + 1$ do
    $j := m$
    while $j > 0$ and $s[j] = t[i + j - 1]$ do $j := j - 1$
    if $j = 0$ then $J \Leftarrow i$
    $i := i + \text{leia\_nihe}(j, t[i + j - 1])$
return $J$
```

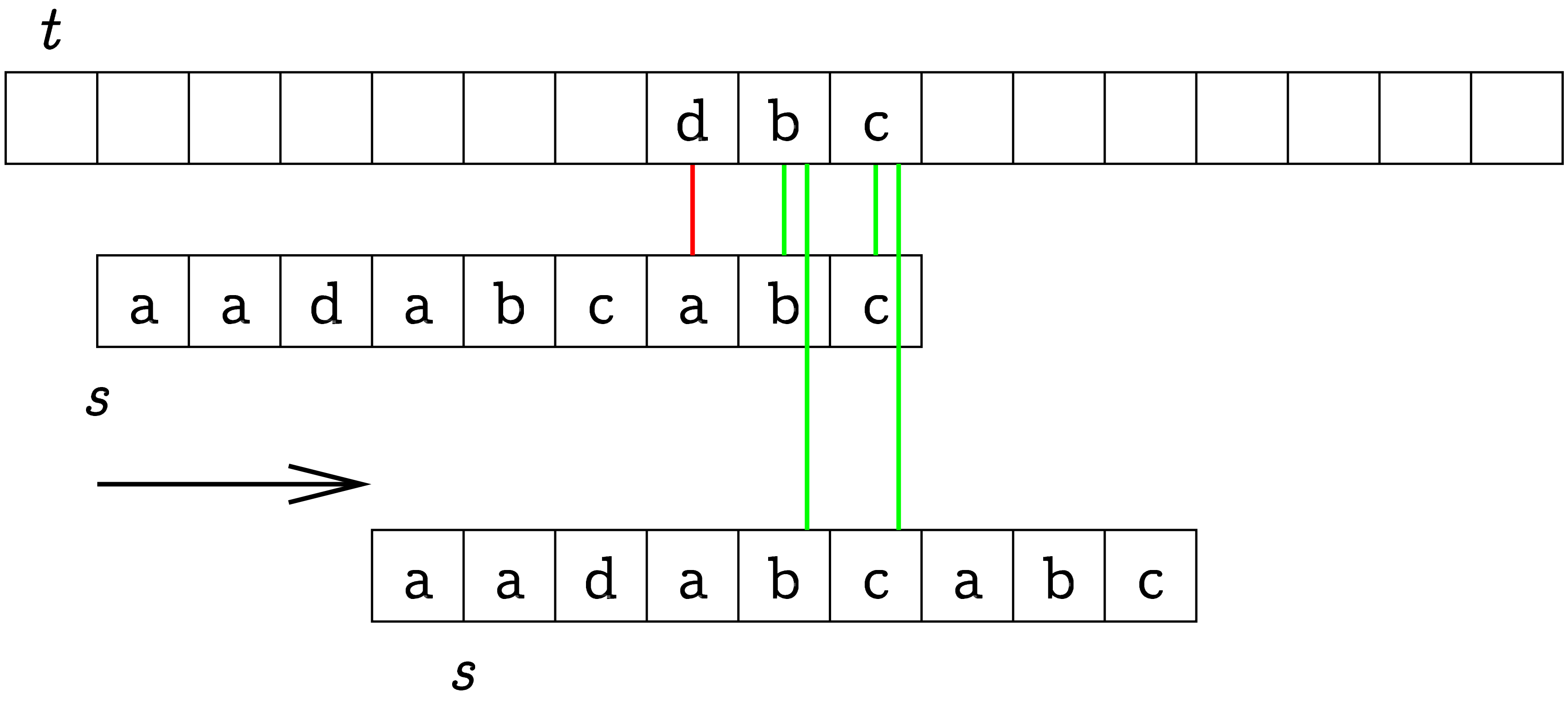
S.t. BM-algoritm käib sõne t lihtsalt vasakult paremale läbi ja uurib, kas seal esineb alamsõnena sõne s. Uurimine käib paremalt vasakule.

Teatavas positsioonis mitteesinemise korral minnakse sõnes t edasi (rida 8). Edasi minnakse vähemalt 1 positsiooni võrra, aga lisaks sellele kasutatakse kahte heuristikat:

ebasobiva tähe heuristika:



sobiva sufiksi heuristika:



Neist kahest heuristikast võetakse parem.

Ebasobiva tähe heuristika kasutamiseks tuleb meil iga $a \in \Sigma$ jaoks leida tema viimane esinemiskoht s-s.

Olgu λ massiiv, mis on indekseeritud Σ elementidega. Siis leia_viimased_kohad(s, Σ) on

```
for all $a \in \Sigma$ do $\lambda_a := 0$
for $i := 1$ to $|s|$ do
    $\lambda_{s[i]} := i$
return $\lambda$
```

Sobiva sufiksi heuristika kasutamiseks: Olgu j see koht, kus tekkis erinevus s ja t vahel.

S.t. meie sobivaks sufiksiks on $s[j+1\ldots m]$.

Meil tuleb leida suurim selline γ_j, et

- $\gamma_j < m$;
- kehtib üks järgmistest väidetest:
  - $s[j+1\ldots m] \sqsupset s[1\ldots\gamma_j]$ (kui $m-j \leqslant \gamma_j$),
  - $s[1\ldots\gamma_j] \sqsupset s[j+1\ldots m]$ (kui $\gamma_j \leqslant m-j$);

γ_j on korrektselt defineeritud, sest 0 rahuldab alati neid kahte tingimust.

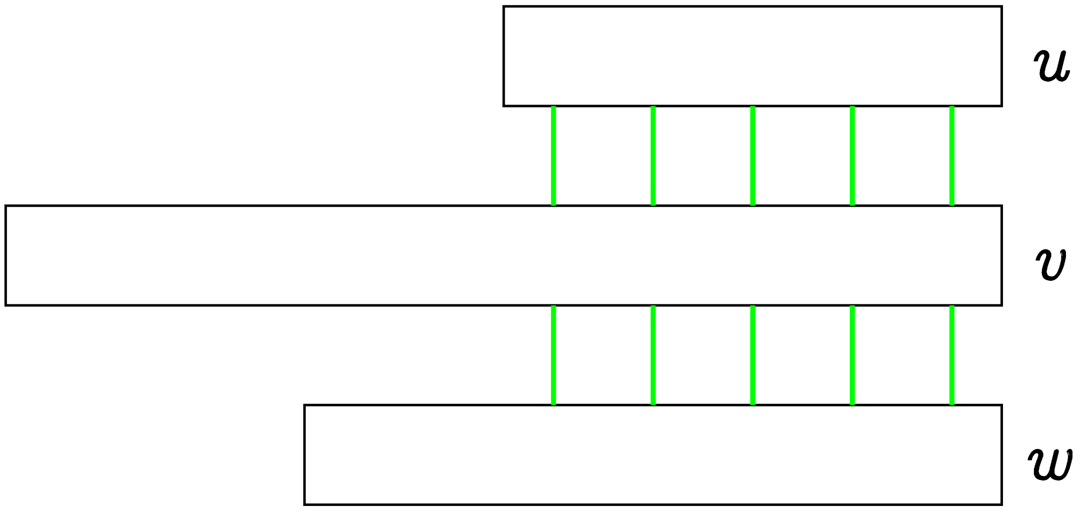
Tähistagu $u \sim v$, kus $u, v \in \Sigma^*$, seda, et $u \sqsupset v$ või $v \sqsupset u$.

Teisisõnu, kui $u \sim v$, siis u ja v lõpud langevad kokku kuni neist kahest lühema sõne alguseni.

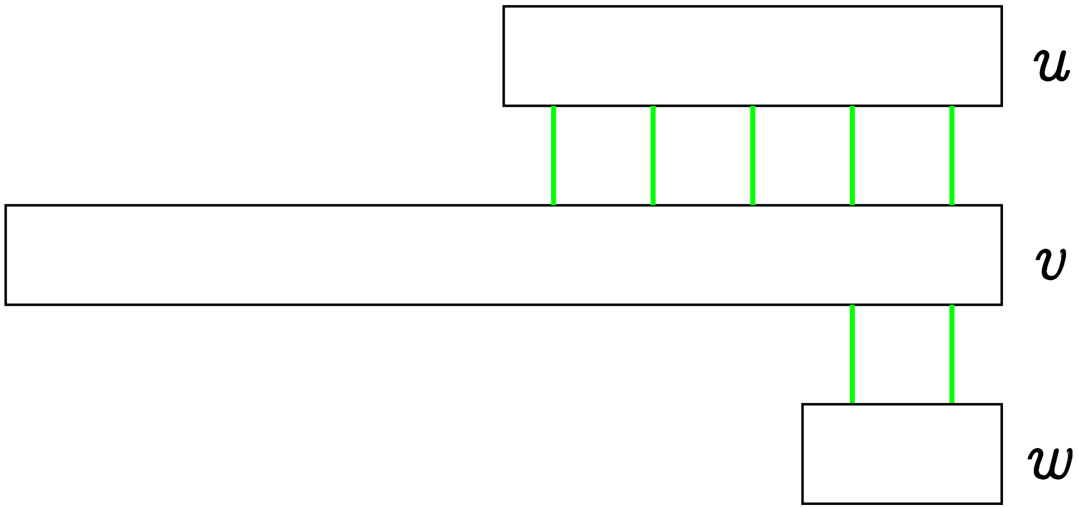
$$\gamma_j := \max\{k \,:\, 0 \leqslant k < m \text{ ja } s[1\ldots k] \sim s[j+1\ldots m]\}$$

Näitame, et kui $u \sim v$ ja $w \sqsupset v$ $(u, v, w \in \Sigma^*)$, siis $u \sim w$.

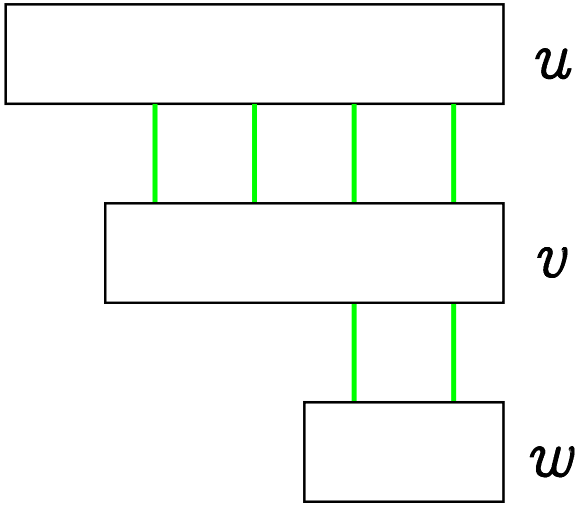
Kui $|u| < |v|$ ja $|u| \leqslant |w|$, siis



Kui $|u| < |v|$ ja $|u| > |w|$, siis



Kui $|u| \geqslant |v|$, siis



Kuna $s[1 \ldots \pi_m] \sim s$ ja $s[j+1 \ldots m] \sqsupset s$, siis $s[1 \ldots \pi_m] \sim s[j+1 \ldots m]$. Seega kuulub π_m hulka, mis esineb γ_j definitsioonis, ning järelikult

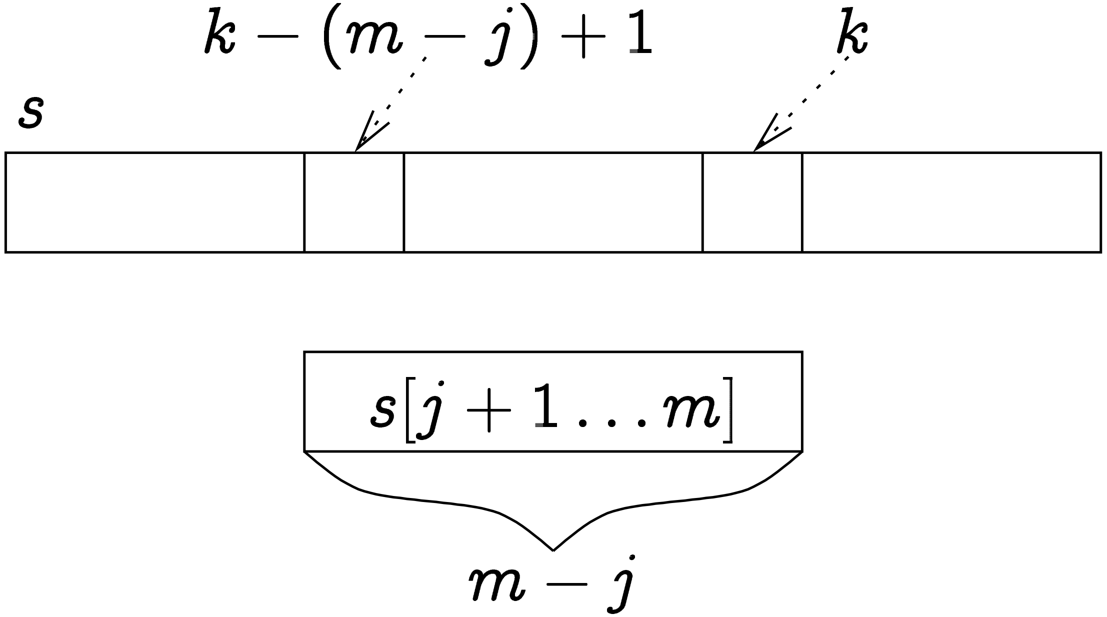
$$\gamma_j = \max\{k : \pi_m \leqslant k < m \text{ ja } s[1 \ldots k] \sim s[j+1 \ldots m]\}$$

Kui $s[1 \ldots k] \sqsupset s[j+1 \ldots m]$, siis $s[1 \ldots k] \sqsupset s$ ja vastavalt π_m definitsioonile $k \leqslant \pi_m$. Seega

$$\gamma_j = \max\bigl(\{\pi_m\} \cup \{k : \pi_m < k < m \text{ ja } s[j{+}1 \ldots m] \sqsupset s[1 \ldots k]\}\bigr).$$

Väide $s[j+1\dots m] \sqsupset s[1\dots k]$ on samaväärne väitega $s[j+1\dots m] \sqsubset s[k-m+j+1\dots m]$.

$$\begin{aligned}\gamma_j = \max\big(\{\pi_m\} \cup \{k : \pi_m < k < m \text{ ja} \\ s[j+1\dots m] \sqsubset s[k-m+j+1\dots m]\}\big)\end{aligned}$$



Olgu π' s-i „sufiksfunktsioon“ (analoogiline prefiksfunktsioonile). S.t. iga $i \in \{1, \ldots, m\}$ jaoks

- $\pi'_i \in \{i+1, i+2, \ldots, m+1\}$;
- $s[\pi'_i \ldots m] \sqsubset s[i \ldots m]$;
- π'_i on vähim arv, mis eelmisi tingimusi rahuldab.

Näiteks kui $s =$ „aabcaabca“, siis π' on:

i	1	2	3	4	5	6	7	8	9
π'_i	5	6	7	8	9	9	10	10	10

Kui $\bar{s}$ on s „paremalt vasakule“ ja $\bar{\pi}$ on tema prefiksfunktsioon, siis kehtib $i + \bar{i} = m + 1 \Rightarrow \pi'_i + \bar{\pi}_{\bar{i}} = m + 1$. Seega $\pi'_i = m + 1 - \bar{\pi}_{m+1-i}$.

Olgu $\pi_m < k < m$. Vaatame tingimust $s[j+1\ldots m] \sqsubset s[k-m+j+1\ldots m]$. Olgu $l = \pi'_{k-m+j+1}$.

- Kui $l > j+1$, siis vaadeldav tingimus ei kehti.
- Kui $l = j+1$, siis vaadeldav tingimus kehtib.
- Kui $l < j+1$, siis võib tingimus kehtida või mitte kehtida. Kui ta aga kehtib, siis ka $s[j+1\ldots m] \sqsubset s[l\ldots m]$.

Viimasel juhul olgu $k' = l - 1 - j + m$. Siis $k' > k$ ja $s[j+1\ldots m] \sqsubset s[k'-m+j+1\ldots m]$. S.t. k polnud maksimaalne selline, mille korral antud tingimus kehtib.

$$\gamma_j = \max\bigl(\{\pi_m\} \cup \{k \,:\, \pi_m < k < m \text{ ja } \pi'_{k-m+j+1} = j+1\}\bigr)$$

```
arvuta_sobiva_sufiksi_pikkus($s$) on

$m := |s|$; $\bar{s} :=$ pööra_ümber($s$)
$\pi :=$ leia_pi($s$); $\bar{\pi} :=$ leia_pi($\bar{s}$)
for $i := 1$ to $m$ do $\pi'_i := m + 1 - \bar{\pi}_{m+1-i}$
for $j := 1$ to $m$ do $\gamma_j := \pi_m$
for $h := 1$ to $m$ do
    $j := \pi'_h - 1$
    $k := h - 1 - j + m$
    if $\gamma_j < k$ then $\gamma_j := k$
return $\gamma$
```

```
initsialiseeri_BM on

$\lambda :=$ leia_viimased_kohad($s$)
$\gamma :=$ arvuta_sobiva_sufiksi_pikkus($s$)
```

```
leia_nihe($j, x$) on

if $j > 0$ then
    $nl := j - \lambda_x$
else
    $nl := 0$
$ng := m - \gamma_j$
return max($nl, ng$)
```