Advanced Algorithms – COMS31900

2013/2014

Lecture 13

Approximate pattern matching (part two)

Benjamin Sach

University of Bristol

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)

	0	1	2	3	4	5	6	7	8	9	10	11	12
T	a	b	c	d	a	b	a	a	d	a	c	a	a
P					a	b	d	a					

n (length of T), m (length of P)

Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \ldots i+m-1]$

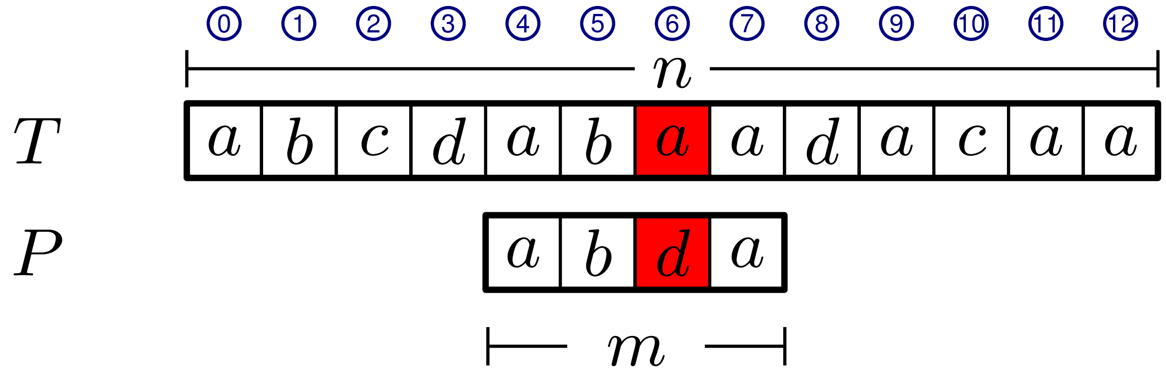
The Hamming distance is the number of (single character) mismatches...

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

University of BRISTOL

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)



$\text{Ham}(4) = 1$

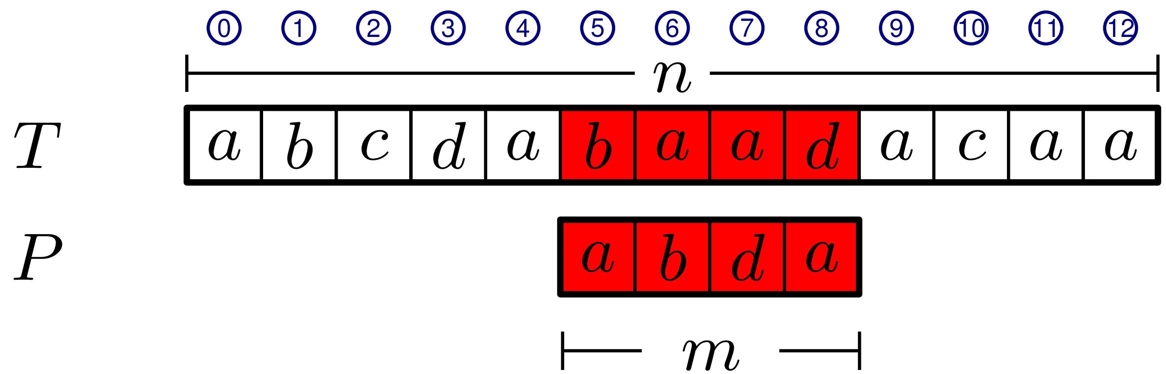
Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \ldots i+m-1]$

The Hamming distance is the number of (single character) mismatches...

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)



$\text{Ham}(5) = 4$

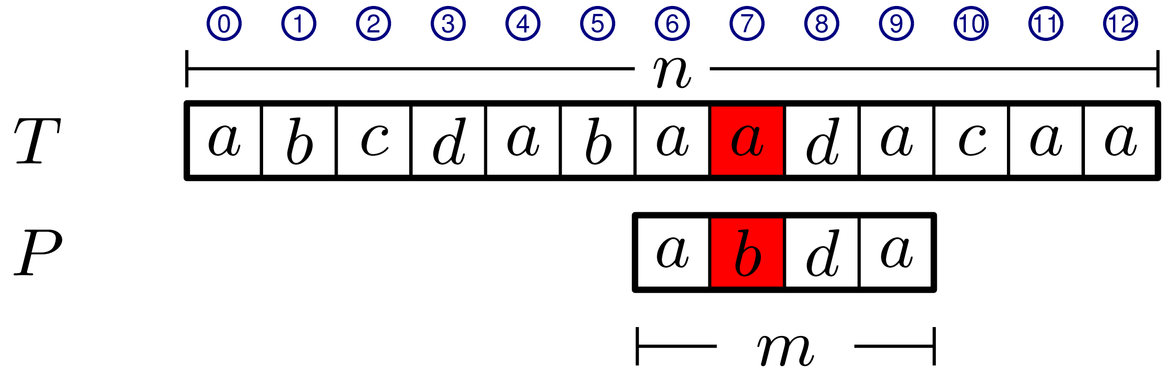
Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \ldots i+m-1]$

The Hamming distance is the number of (single character) mismatches. . .

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)



$\text{Ham}(6) = 1$

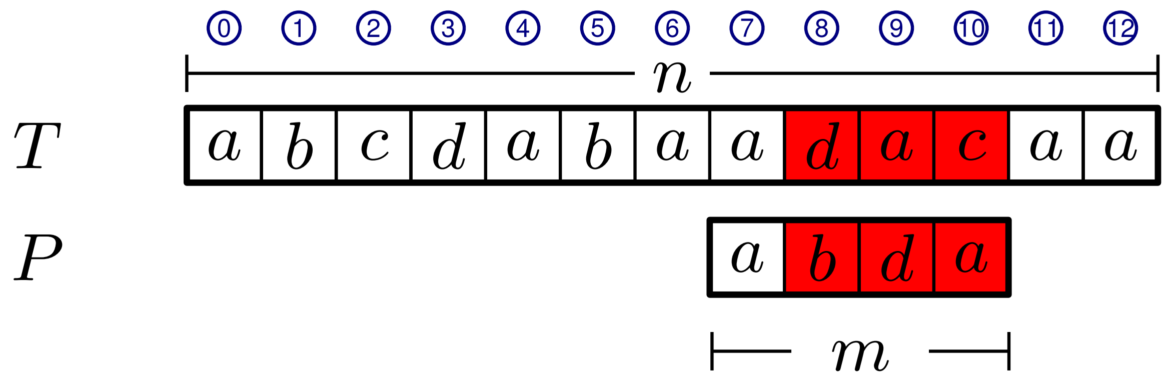
Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \dots i+m-1]$

The Hamming distance is the number of (single character) mismatches...

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)



$\text{Ham}(7) = 3$

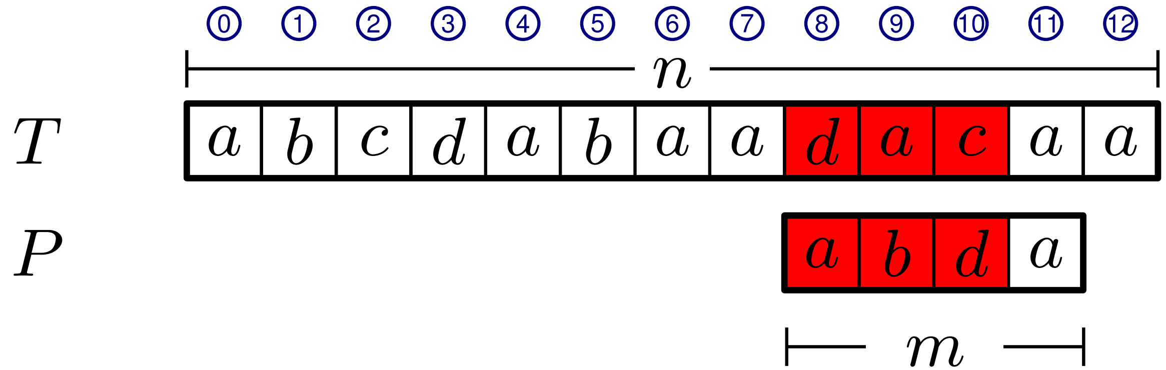
Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \ldots i+m-1]$

The Hamming distance is the number of (single character) mismatches. . .

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

Pattern matching with mismatches (Hamming distance)

Input A text string T (length n) and a pattern string P (length m)



$\text{Ham}(8) = 3$

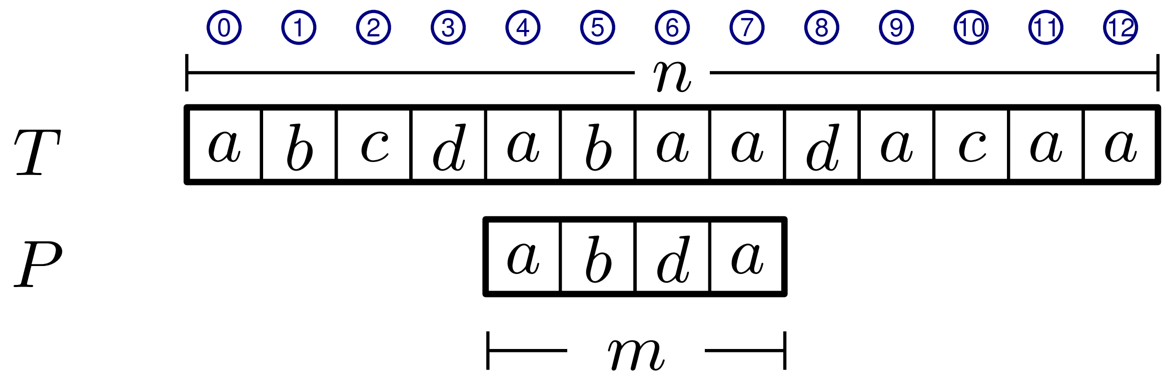
Goal: For all i, output, Ham(i), the Hamming distance between P and $T[i \ldots i+m-1]$

The Hamming distance is the number of (single character) mismatches. . .

i.e. the number of distinct j such that $P[j] \neq T[i+j]$

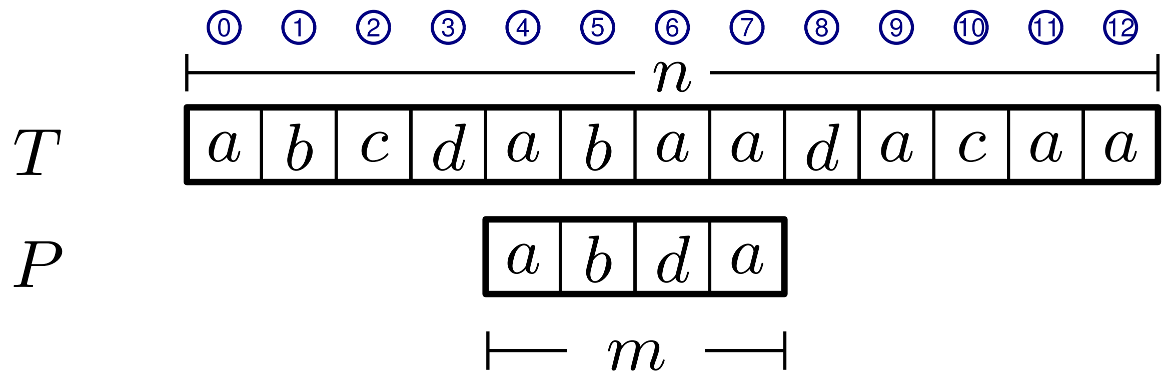
Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .



Hamming distance - considering symbols seperately

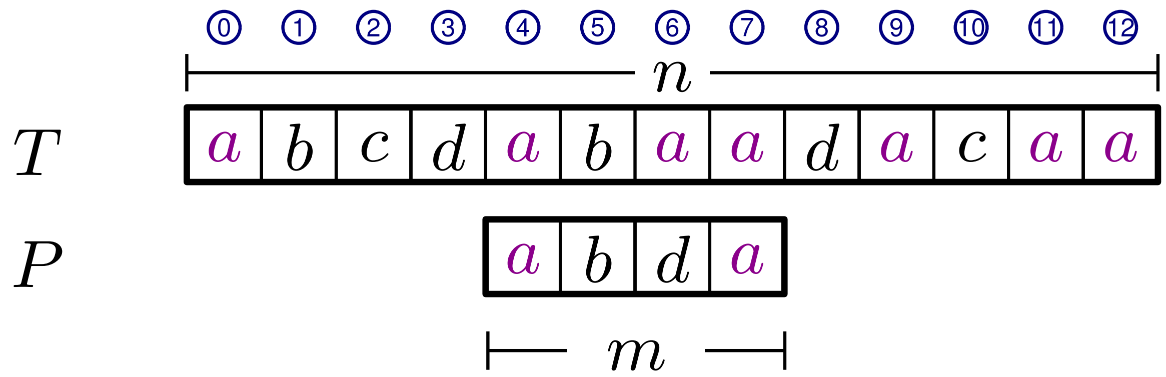
Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .



Replace all a *symbols with* 1 *and everything else with* 0

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually...



Replace all a *symbols with* 1 *and everything else with* 0

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .

n

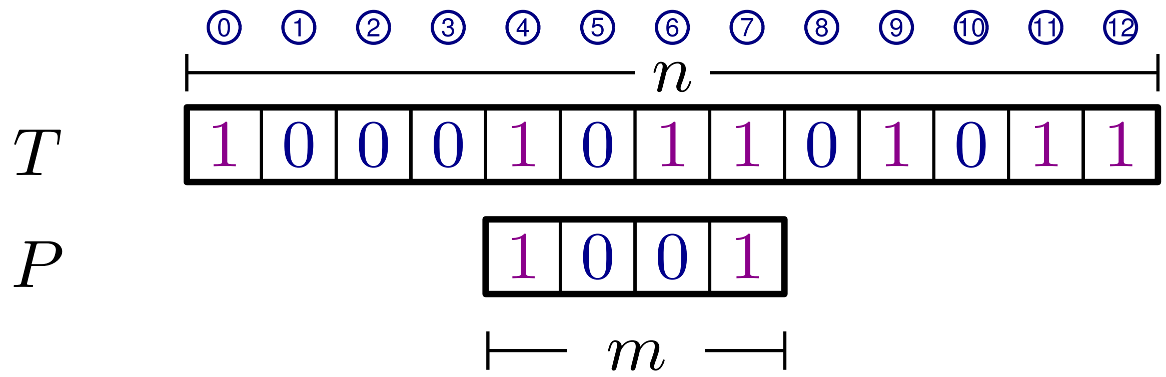
	0	1	2	3	4	5	6	7	8	9	10	11	12
T	1	b	c	d	1	b	1	1	d	1	c	1	1
P					1	b	d	1					

m

Replace all a symbols with 1 and everything else with 0

Hamming distance - considering symbols seperately

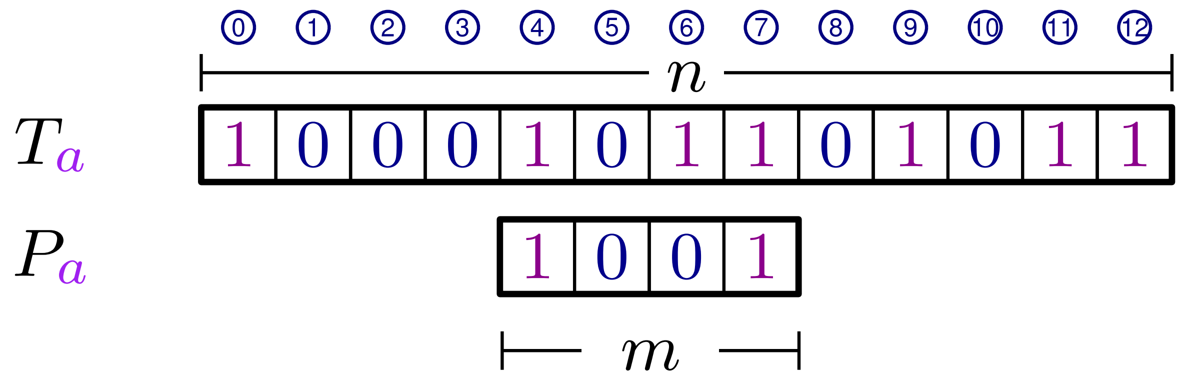
Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .



Replace all a *symbols with* 1 *and everything else with* 0

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually...

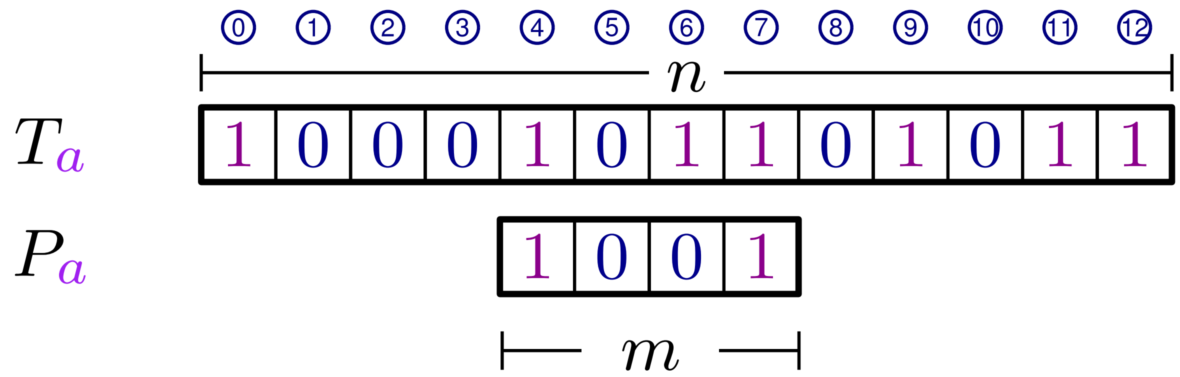


Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider...

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually...



Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider...

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually...

n

	0	1	2	3	4	5	6	7	8	9	10	11	12
T_a	1	0	0	0	1	0	1	1	0	1	0	1	1
P_a					1	0	0	1					

m

Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider...

$$(T_a \otimes P_a)[i] = \sum_{j=0}^{m-1} \underbrace{P_a[j]T_a[i+j]}_{1 \text{ iff } P[j]=T[i+j]=a}$$

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .

n

	0	1	2	3	4	5	6	7	8	9	10	11	12
T_a	1	0	0	0	1	0	1	1	0	1	0	1	1
P_a					1	0	0	1					

m

Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider. . .

$$(T_a \otimes P_a)[i] = \sum_{j=0}^{m-1} \underbrace{P_a[j]T_a[i+j]}_{1 \text{ iff } P[j]=T[i+j]=a}$$

$\cdots \; 2 \; 1 \; 2 \; \boxed{1} \; \boxed{2} \; \boxed{1} \; 3 \; \cdots$

$\times + \times + \times$

$\boxed{1} \; \boxed{2} \; \boxed{1}$

$$(1 \times 1) + (2 \times 2) + (1 \times 1) = 6$$

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually...

	0	1	2	3	4	5	6	7	8	9	10	11	12
	n												
T_a	1	0	0	0	1	0	1	1	0	1	0	1	1
P_a					1	0	0	1					
					m								

Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider...

$$(T_a \otimes P_a)[i] = \sum_{j=0}^{m-1} \underbrace{P_a[j]T_a[i+j]}_{1 \text{ iff } P[j]=T[i+j]=a}$$

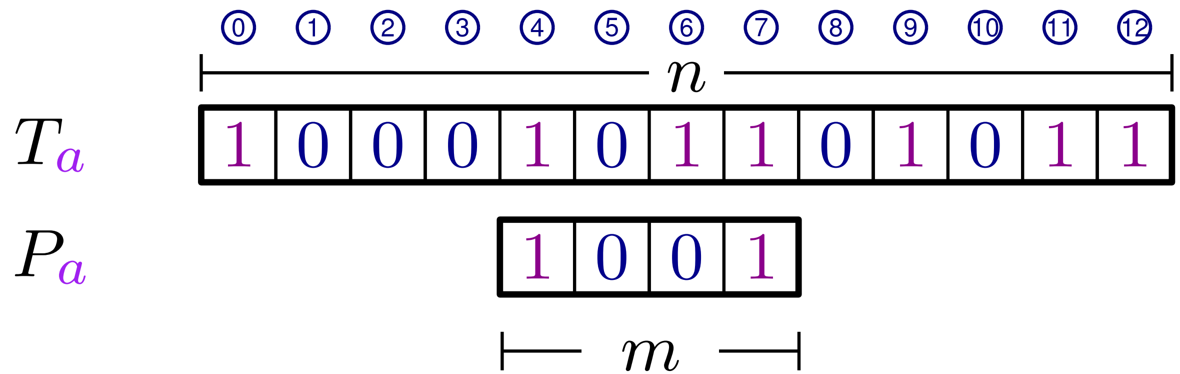
···	2	1	2	1	2	1	3	···
				× +	× +	×		
				1	2	1		

$$(1 \times 1) + (2 \times 2) + (1 \times 1) = 6$$

This is the number of matching as at the i-th alignment.

Hamming distance - considering symbols seperately

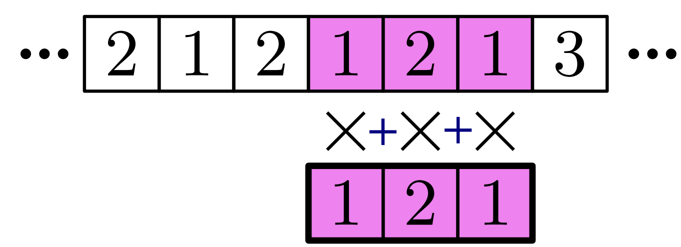
Imagine that the alphabet contains only a small number of different symbols, which we consider individually...



Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider...

$$(T_a \otimes P_a)[i] = \sum_{j=0}^{m-1} \underbrace{P_a[j]T_a[i+j]}_{1 \text{ iff } P[j]=T[i+j]=a}$$



$$(1 \times 1) + (2 \times 2) + (1 \times 1) = 6$$

This is the number of matching as at the i-th alignment.

Hamming distance - considering symbols seperately

Imagine that the alphabet contains only a small number of different symbols, which we consider individually. . .

	0	1	2	3	4	5	6	7	8	9	10	11	12
T_a	1	0	0	0	1	0	1	1	0	1	0	1	1
P_a					1	0	0	1					

n (length of T_a), m (length of P_a)

Replace all a symbols with 1 and everything else with 0

We denote these new strings T_a and P_a and consider. . .

$$(T_a \otimes P_a)[i] = \sum_{j=0}^{m-1} \underbrace{P_a[j]T_a[i+j]}_{1 \text{ iff } P[j]=T[i+j]=a}$$

$$\cdots \; \boxed{2}\boxed{1}\boxed{2}\boxed{1}\boxed{2}\boxed{1}\boxed{3} \; \cdots$$

$$\times + \times + \times$$

$$\boxed{1}\boxed{2}\boxed{1}$$

$$(1 \times 1) + (2 \times 2) + (1 \times 1) = 6$$

This is the number of matching as at the i-th alignment.

which we can compute (for all i) in $O(n \log m)$ time via cross-correlations

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i] \,.$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} \underbrace{(T_\sigma \otimes P_\sigma)[i]}_{\text{matches involving } \sigma}.$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \underbrace{\sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i]}_{\text{all matches}}\,.$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i]\,.$$

mismatches $= m-$ matches

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i]\,.$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i] \, .$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ) ($O(n|\Sigma| \log m)$ time)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i] \,.$$

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ) ($O(n|\Sigma| \log m)$ time)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i]\,.$$ ($O(n|\Sigma|)$ time)

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ) ($O(n|\Sigma| \log m)$ time)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i]\,. \quad (O(n|\Sigma|) \text{ time})$$

This takes $O(n|\Sigma| \log m)$ total time (and $O(n)$ space)

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ) ($O(n|\Sigma| \log m)$ time)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i] \,. \quad (O(n|\Sigma|) \text{ time})$$

This takes $O(n|\Sigma| \log m)$ total time (and $O(n)$ space)

However, $|\Sigma|$ could be as big as m...

Hamming distance - considering symbols seperately

We saw how to find all matches with a single symbol in $O(n \log m)$ time

Let Σ denote the set of alphabet symbols and $|\Sigma|$ be its size

Algorithm Summary

Construct T_σ and P_σ for every symbol σ in Σ ($O(n|\Sigma|)$ time)

Compute $T_\sigma \otimes P_\sigma$ (for every symbol σ in Σ) ($O(n|\Sigma| \log m)$ time)

For every i, compute,

$$\text{Ham}(i) = m - \sum_{\sigma \in \Sigma} (T_\sigma \otimes P_\sigma)[i] \,.$$ ($O(n|\Sigma|)$ time)

This takes $O(n|\Sigma| \log m)$ total time (and $O(n)$ space)

However, $|\Sigma|$ could be as big as m...

what should we do instead?

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

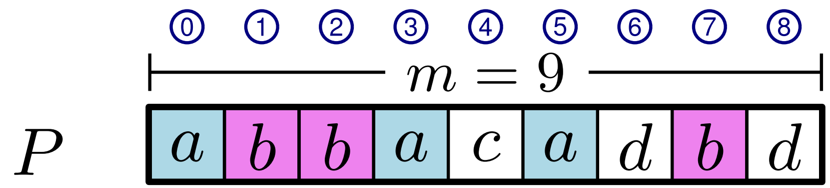
$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

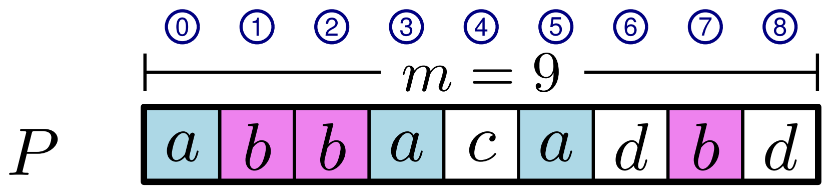


a is *frequent*, b is *frequent*

c and d are not *frequent*

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

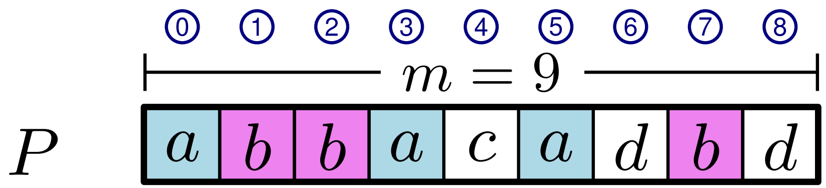


a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.



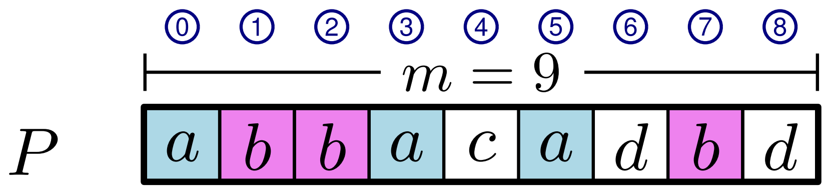
a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.



a is *frequent*, b is *frequent*
c and d are not *frequent*

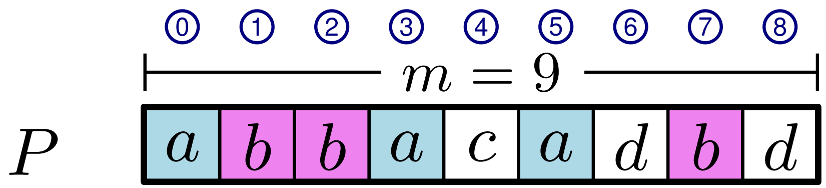
Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.



a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m} + 1)$ freq. symbols

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m} + 1)$ freq. symbols

each occurs at least $\sqrt{m}$ times. . .

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*

c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m}+1)$ freq. symbols

each occurs at least $\sqrt{m}$ times... $(\sqrt{m}+1)\sqrt{m} > m$

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*

c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m}+1)$ freq. symbols

each occurs at least $\sqrt{m}$ times... $(\sqrt{m}+1)\sqrt{m} > m$ Contradiction!

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*
c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m}+1)$ freq. symbols

each occurs at least $\sqrt{m}$ times... $(\sqrt{m}+1)\sqrt{m} > m$ Contradiction!

so there are at most $\sqrt{m}$ *frequent symbols*

The frequent/infrequent symbols trick

Definition: A symbol is *frequent* if it occurs at least $\sqrt{m}$ times in P.

$m = 9$

	0	1	2	3	4	5	6	7	8
P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*

c and d are not *frequent*

Step 1: Count all matches involving frequent symbols.

Consider each frequent symbol separately in $O(n \log m)$ time (per symbol).

using cross-correlations

How many frequent symbols can there be?

Assume that there at least $(\sqrt{m} + 1)$ freq. symbols

each occurs at least $\sqrt{m}$ times... $(\sqrt{m} + 1)\sqrt{m} > m$ Contradiction!

so there are at most $\sqrt{m}$ frequent symbols

So Step 1 takes $O(n\sqrt{m} \log m)$ time.

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

Every symbol is either frequent or infrequent

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							

$m = 9$

a is *frequent*, b is *frequent*
c and d are *infrequent*

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							

$m = 9$

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

Every symbol is either frequent or infrequent

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	0	0	0	0	0	0	0	0

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n-m+1)$ - which is initially all zeros

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	0	0	0	0	0	0	0	0

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*

c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T \ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	a	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	$\cancel{a}$	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	$\cancel{a}$	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

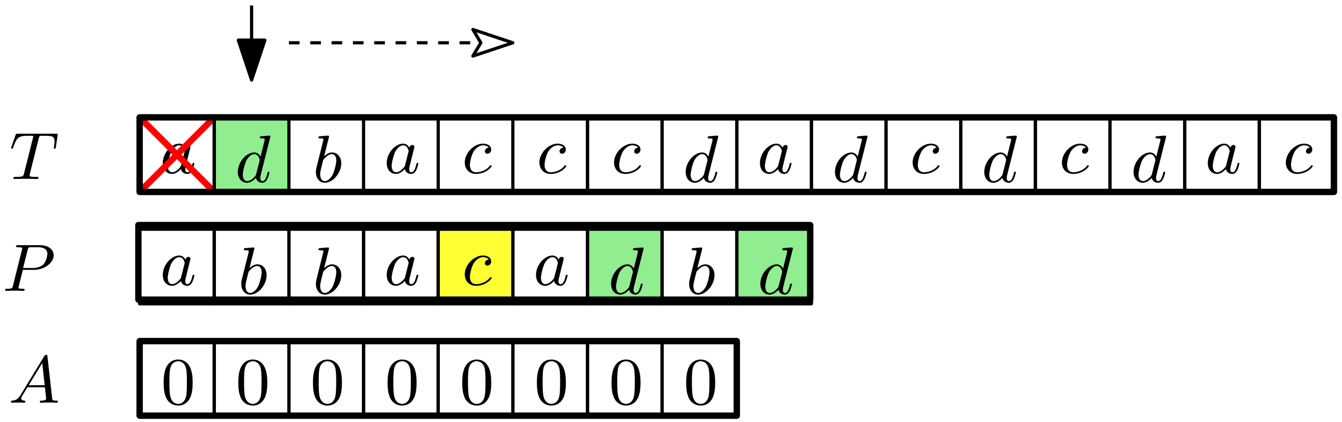
Make a single pass through $T \ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	$\cancel{a}$	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	b	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

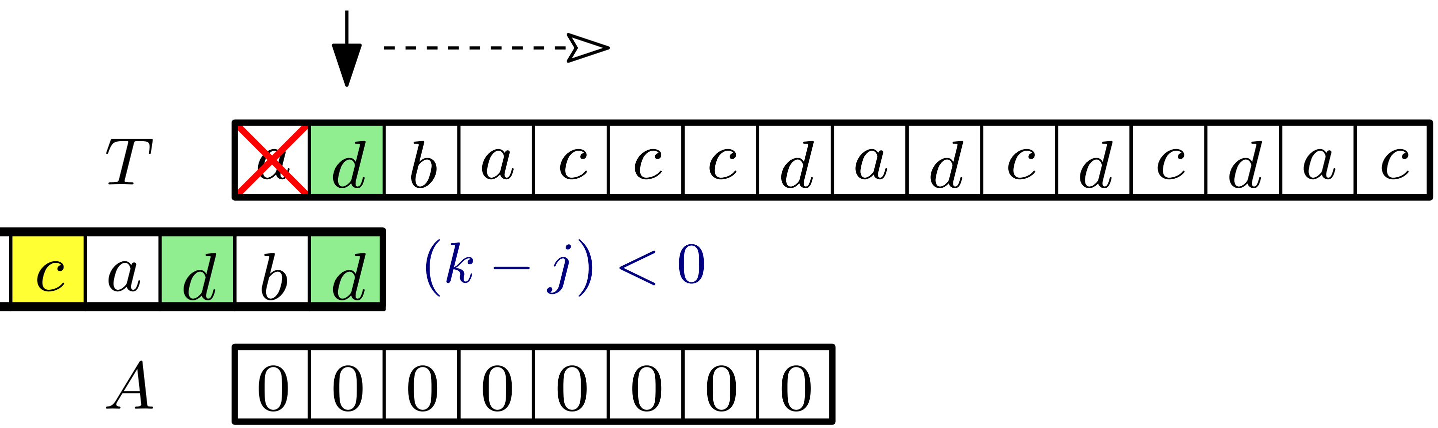
For all j such that $T[k] = P[j]$,

Increase $A[k-j]$ by one

except when $(k-j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

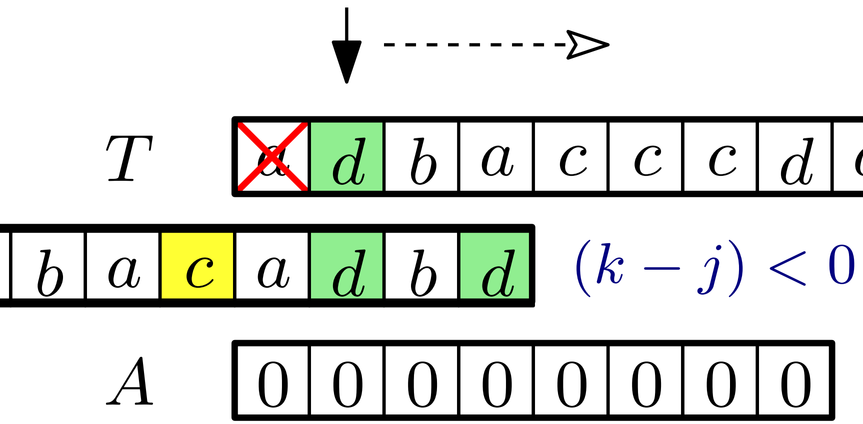
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n-m+1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

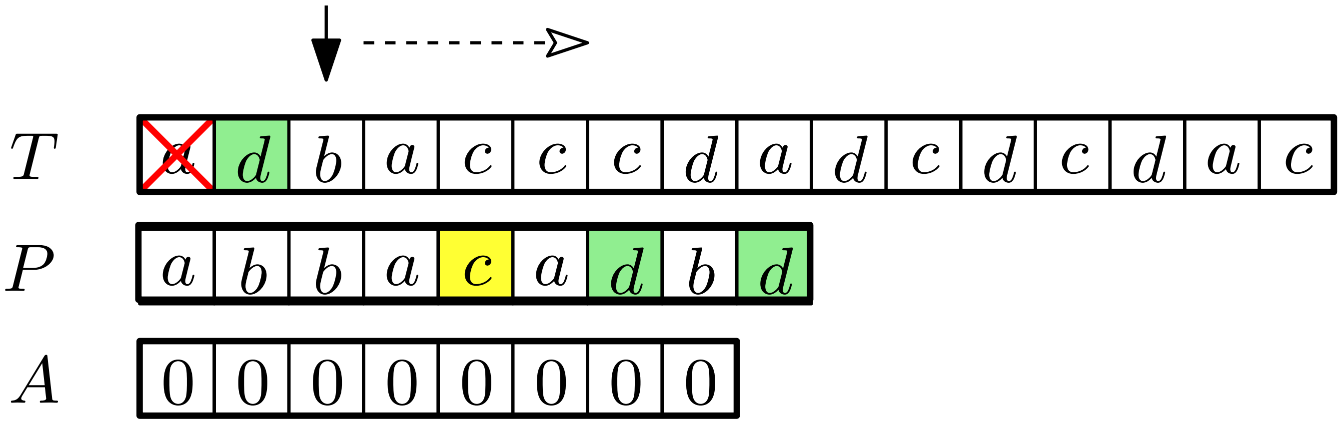
For all j such that $T[k] = P[j]$,

Increase $A[k-j]$ by one

except when $(k-j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

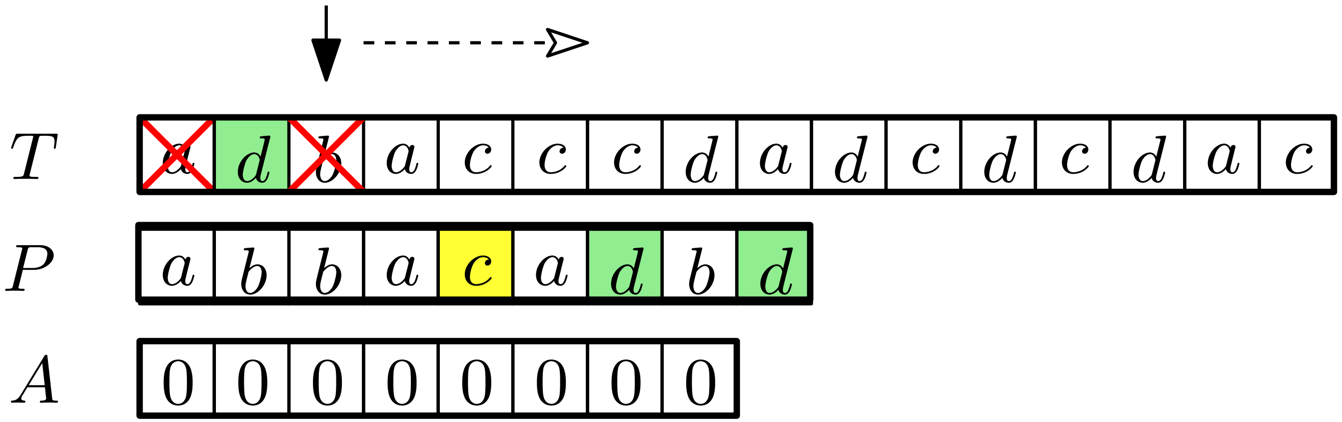
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	~~b~~	a	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	$\cancel{a}$	d	$\cancel{b}$	$\cancel{a}$	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

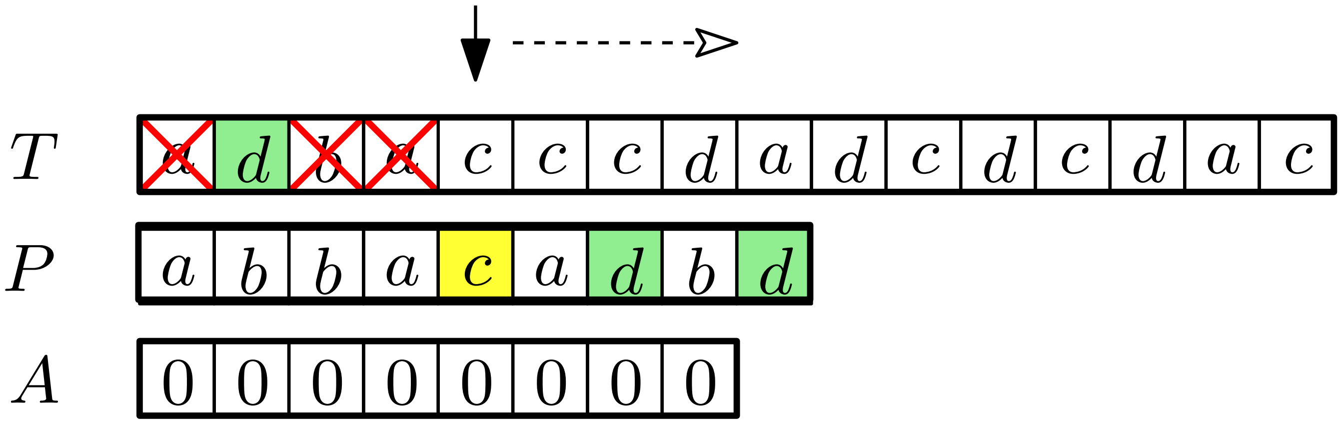
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	~~b~~	~~a~~	c	c	c	d	a	d	c	d	c	d	a	c
P	a	b	b	a	c	a	d	b	d							
A	0	0	0	0	0	0	0	0								

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

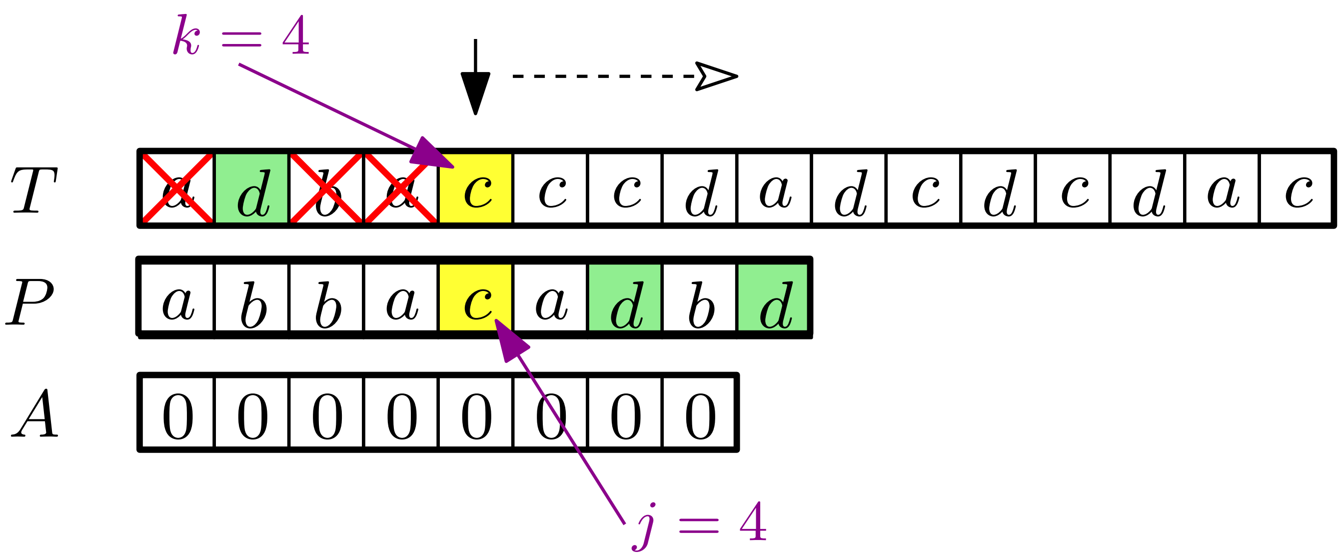
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

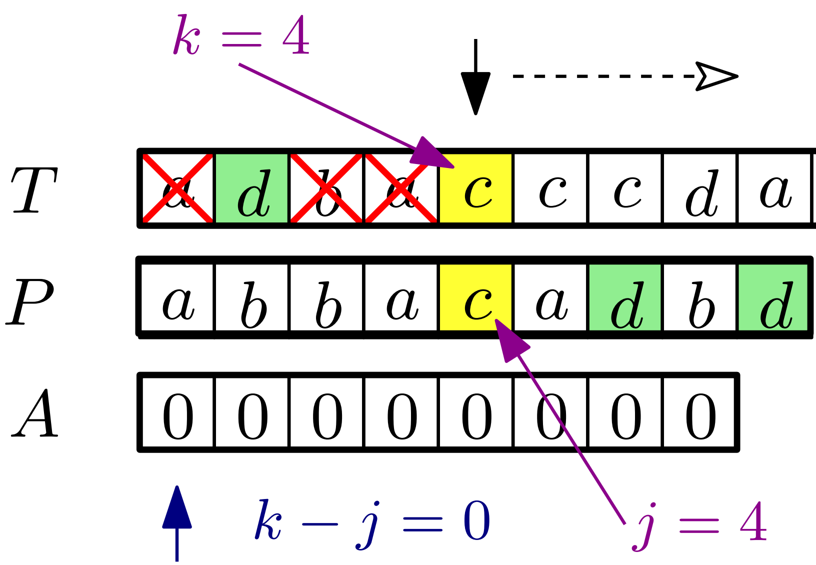
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

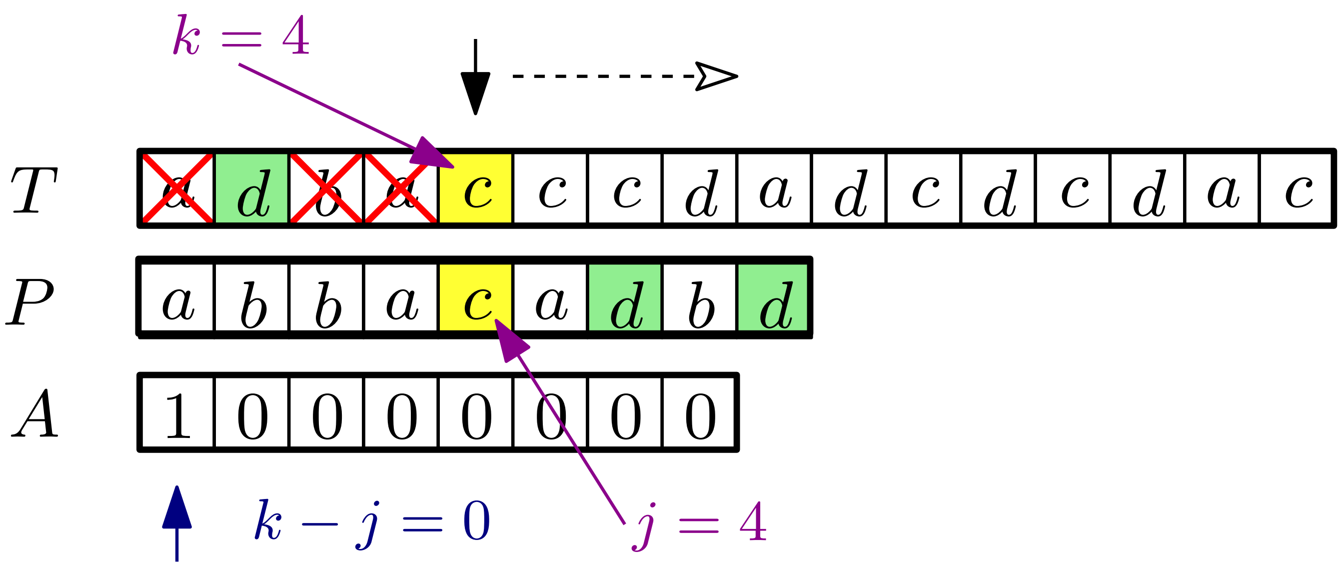
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

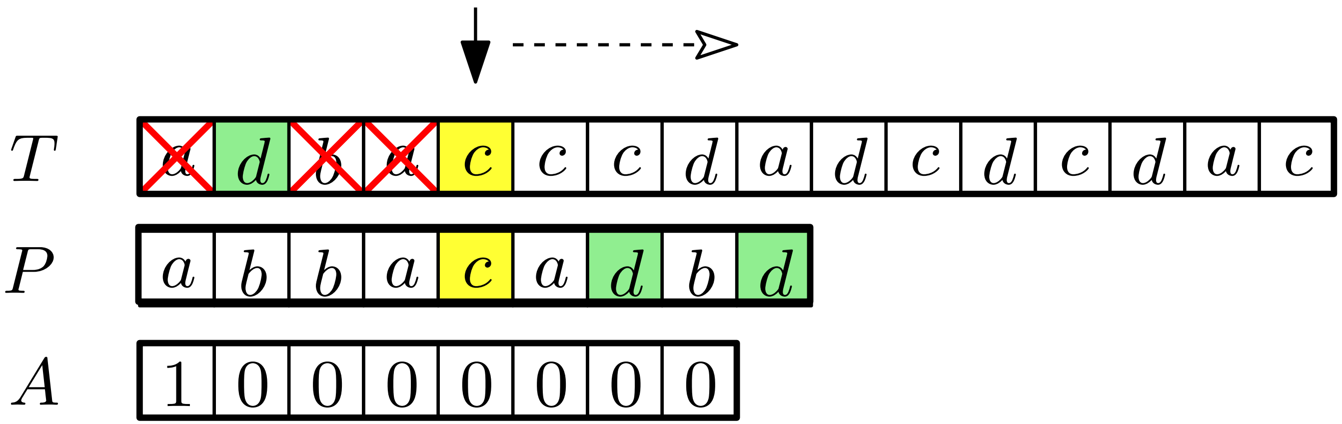
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

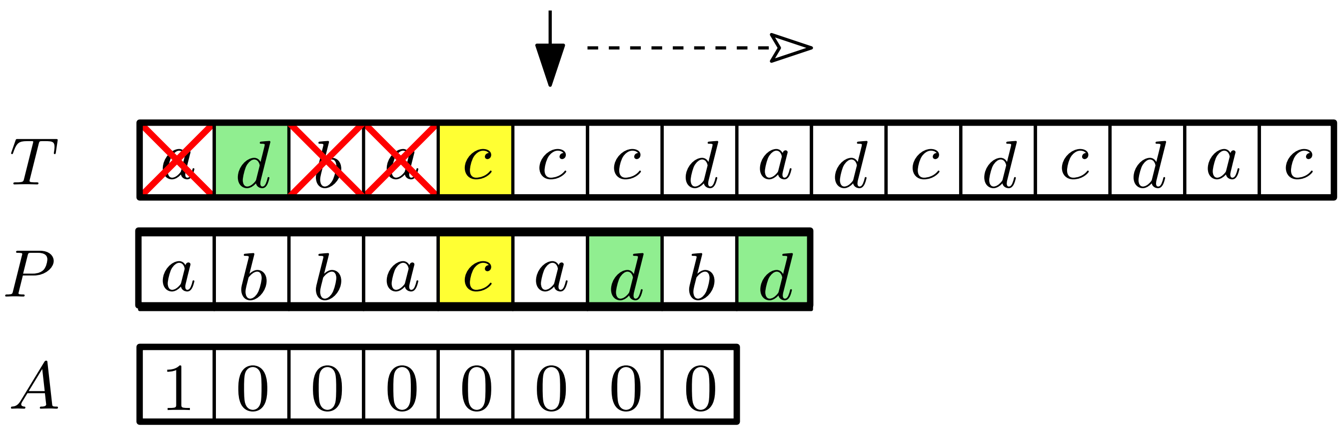
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

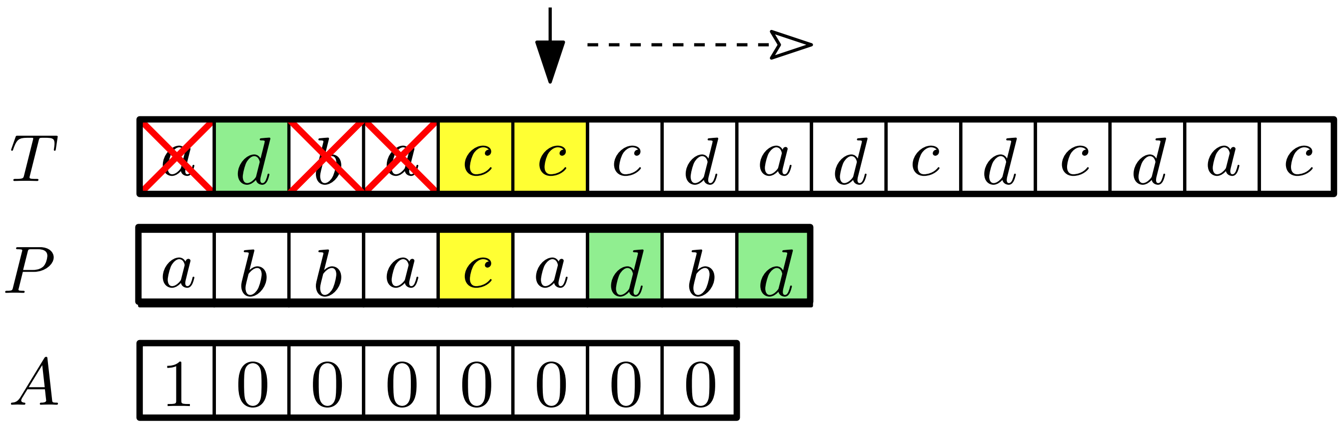
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

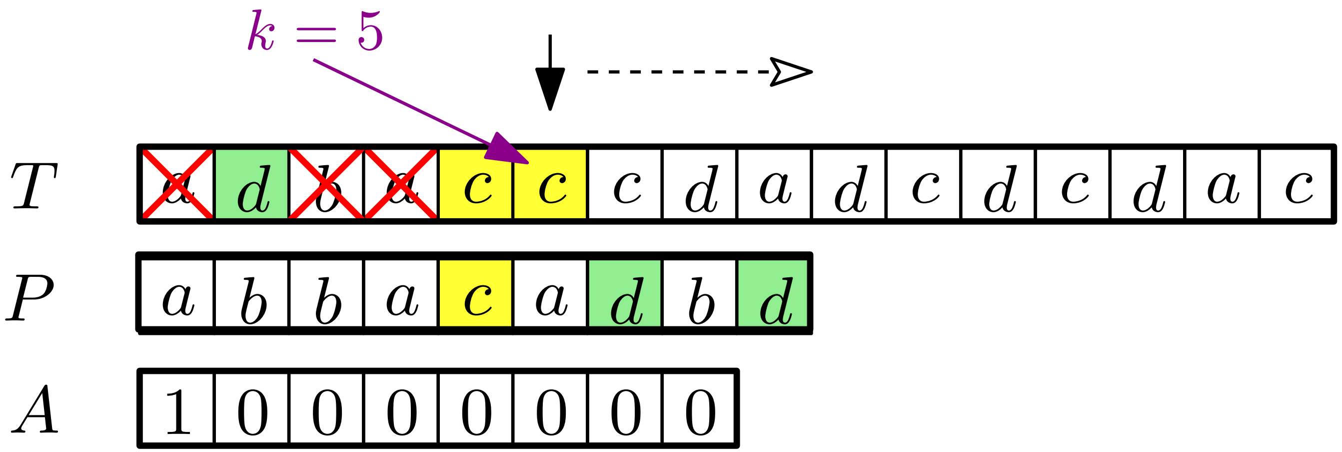
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

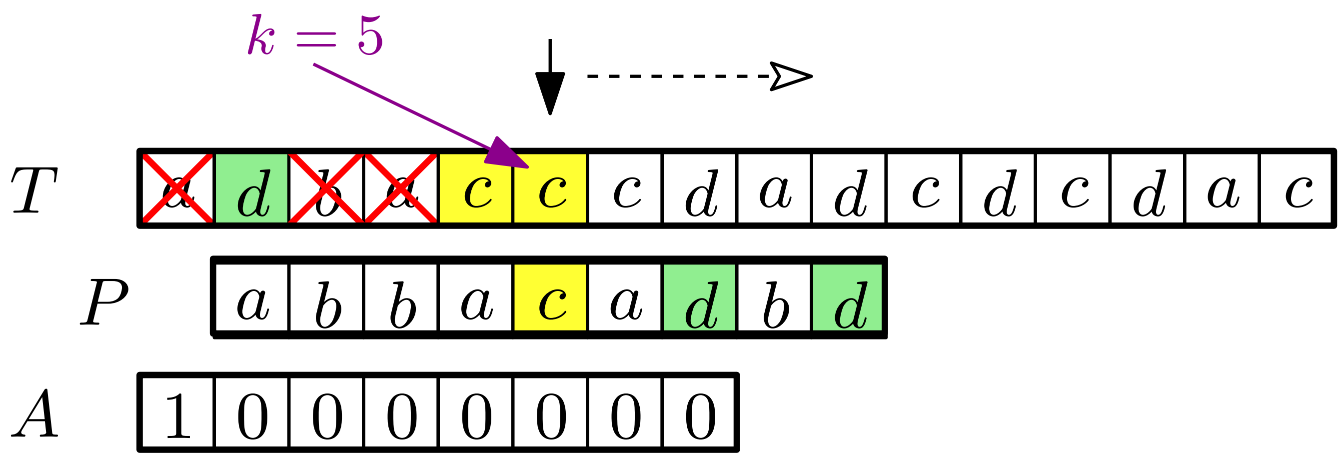
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

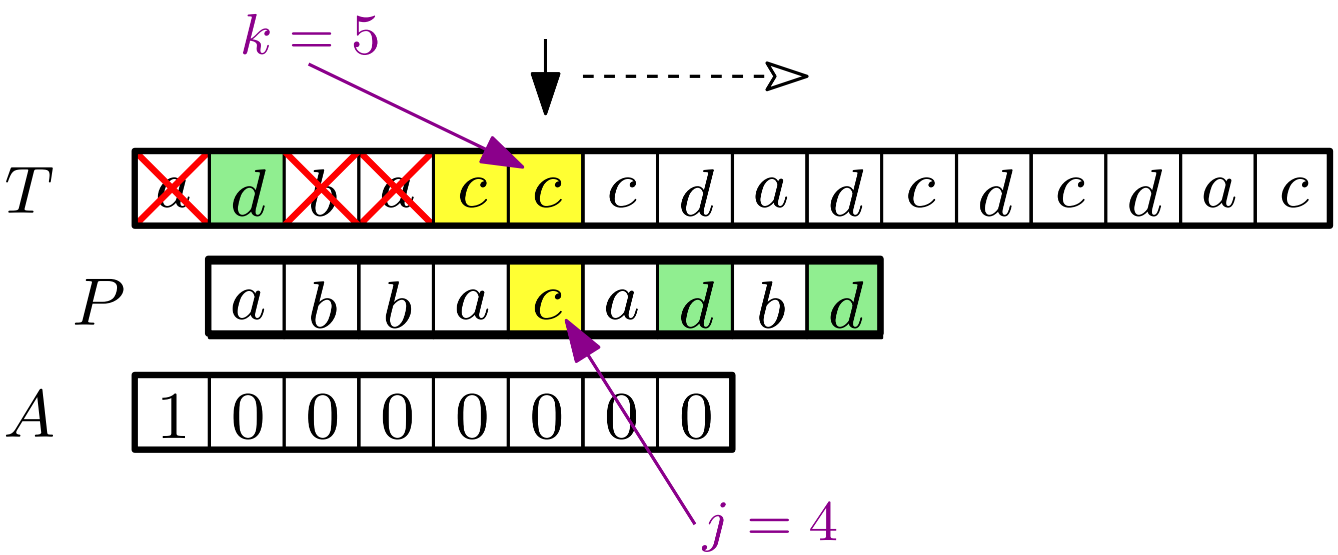
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

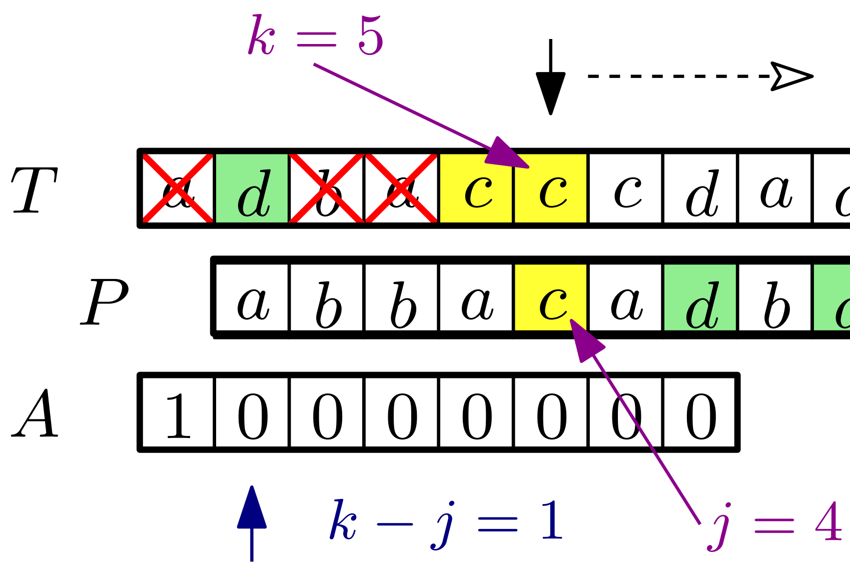
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

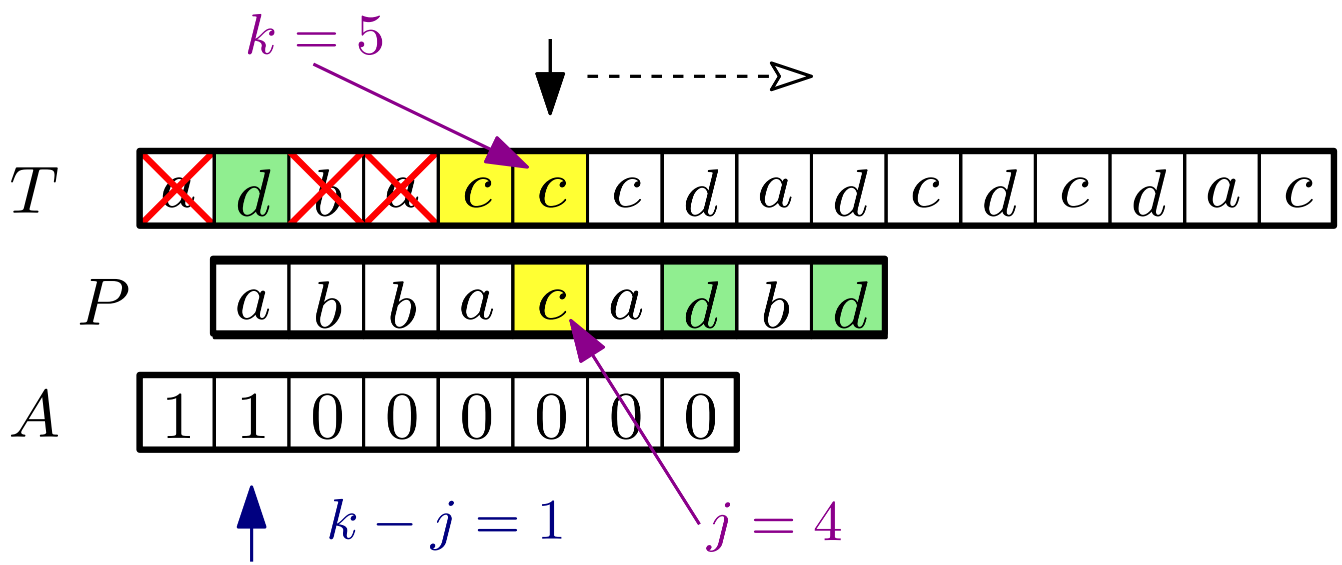
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

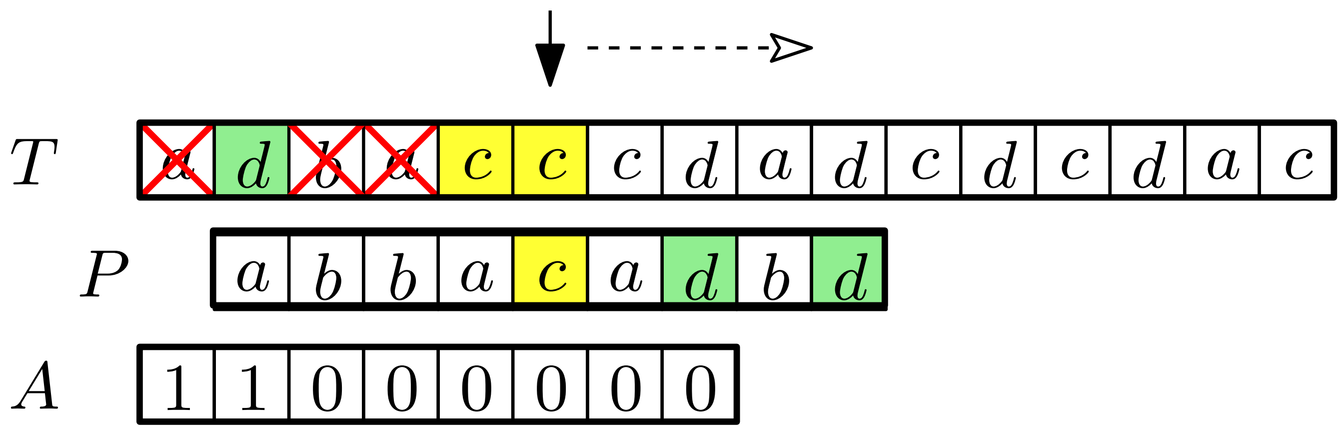
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

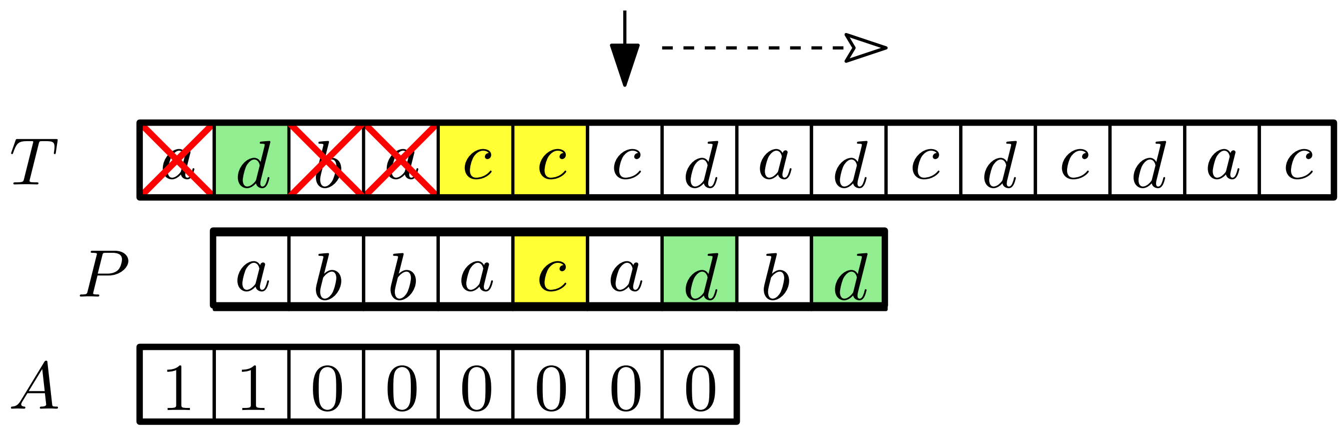
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

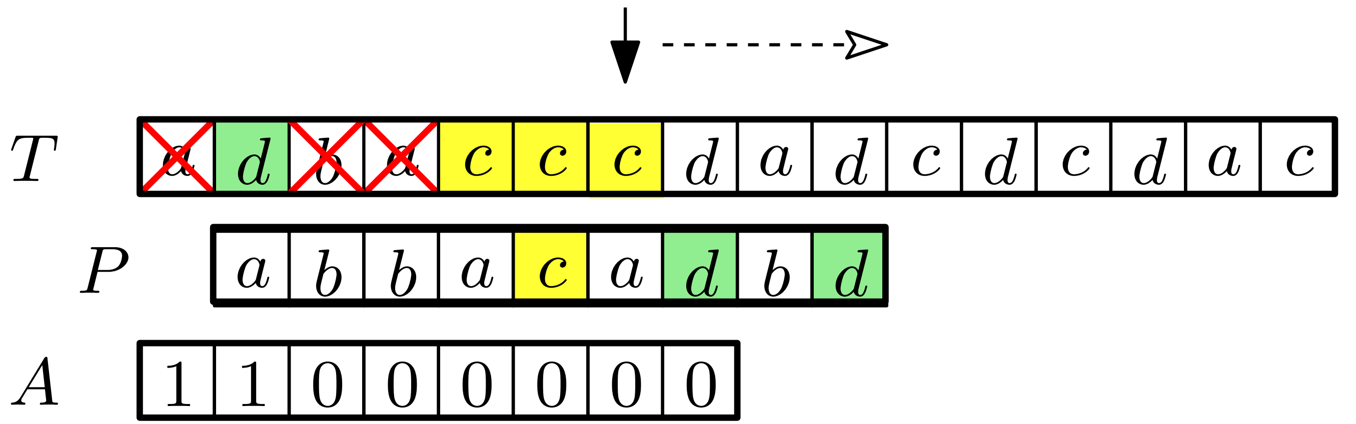
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

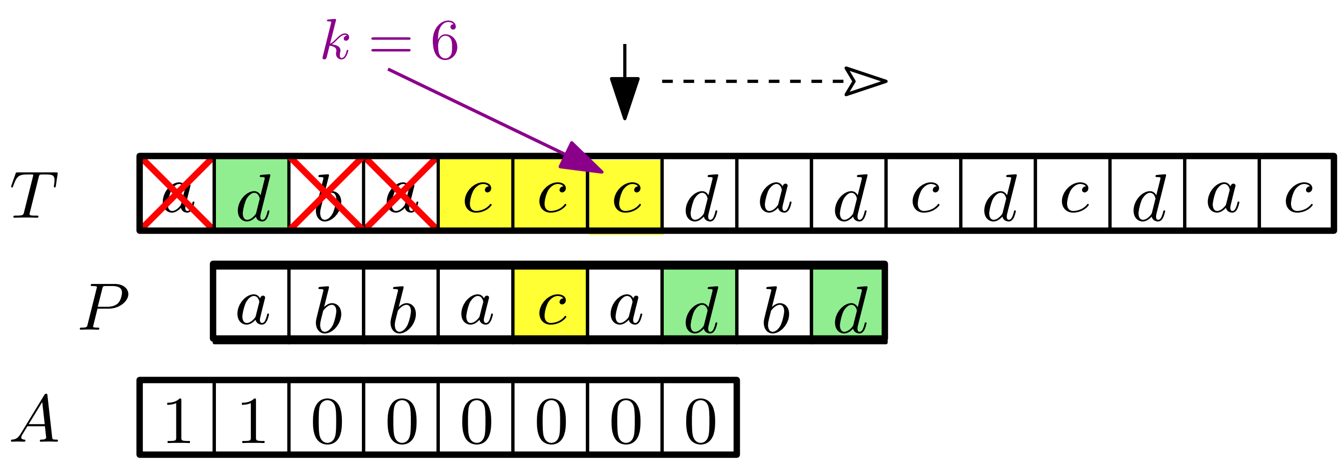
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

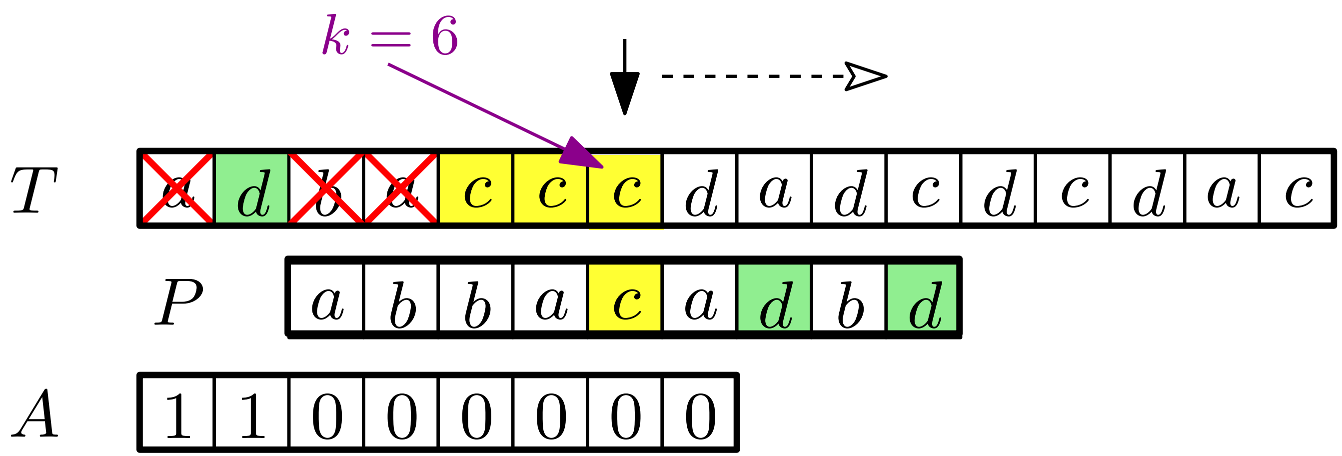
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

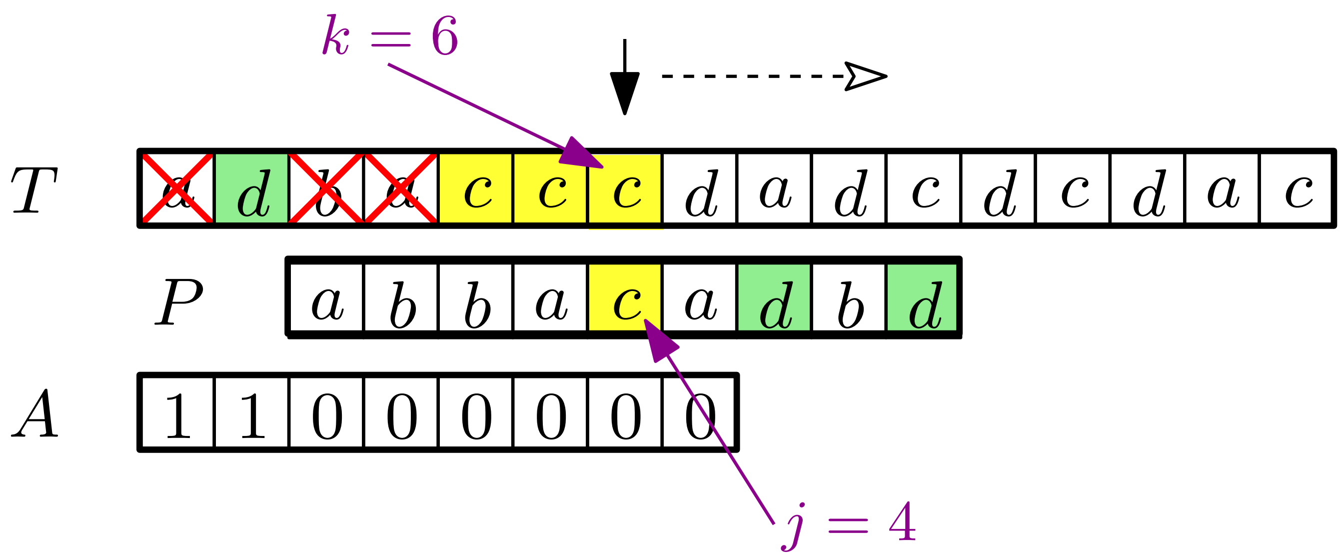
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

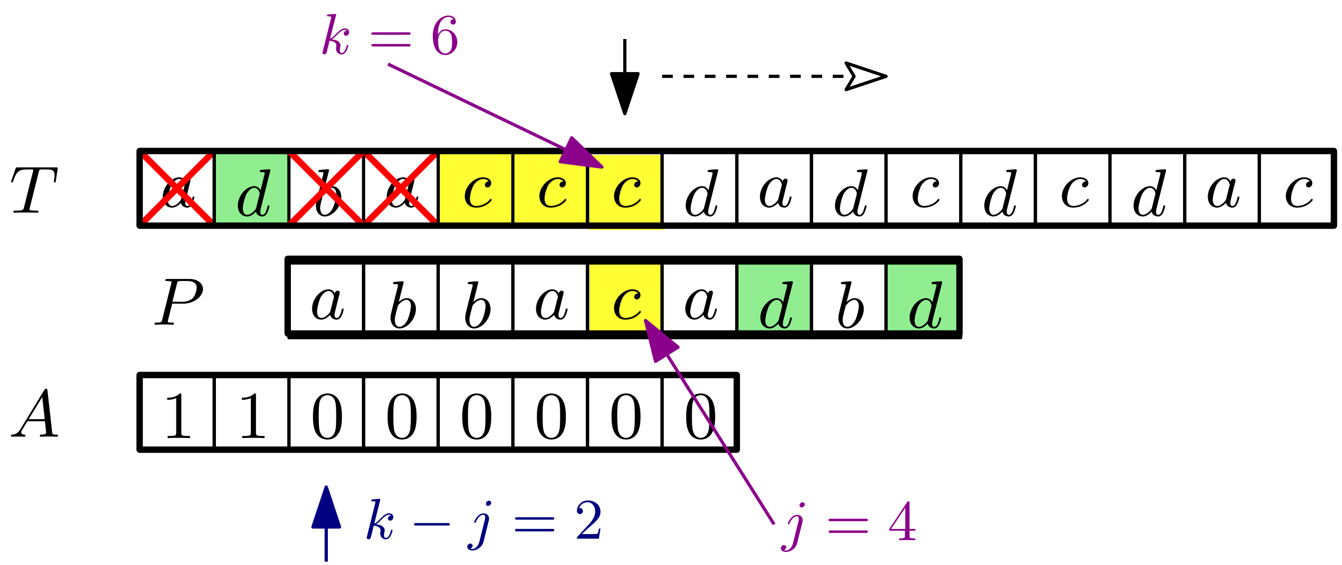
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

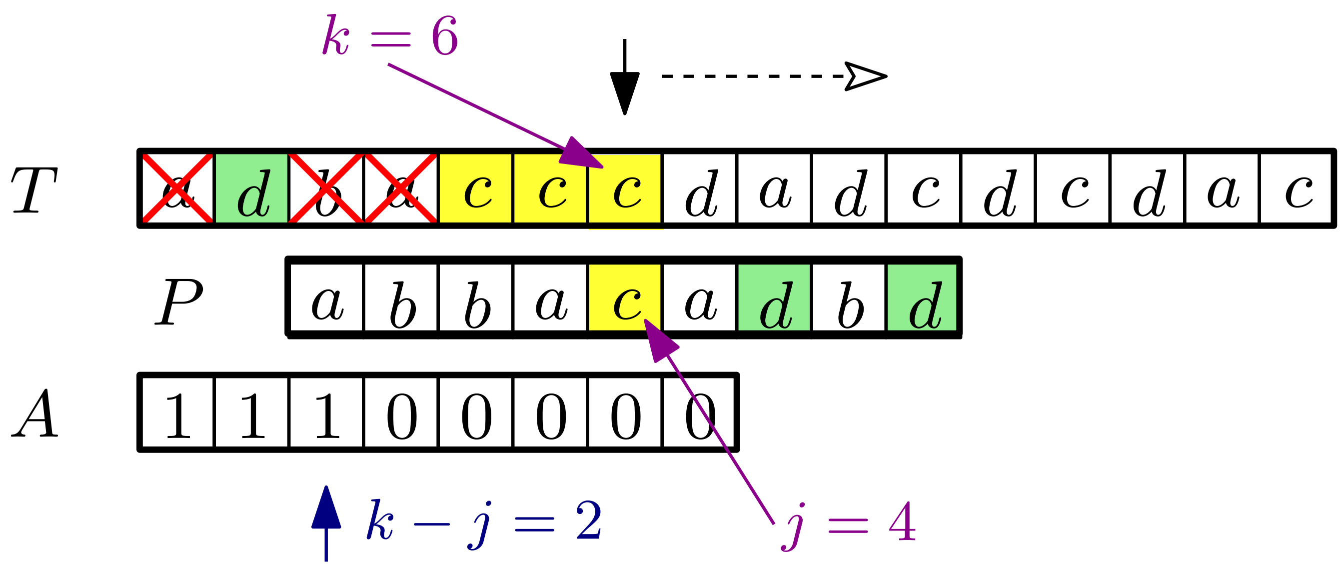
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

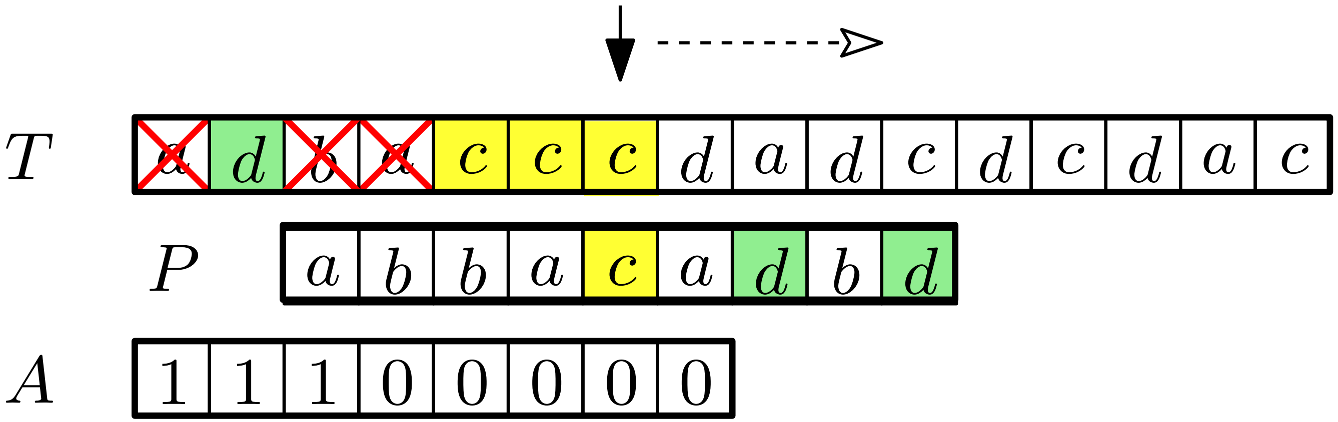
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

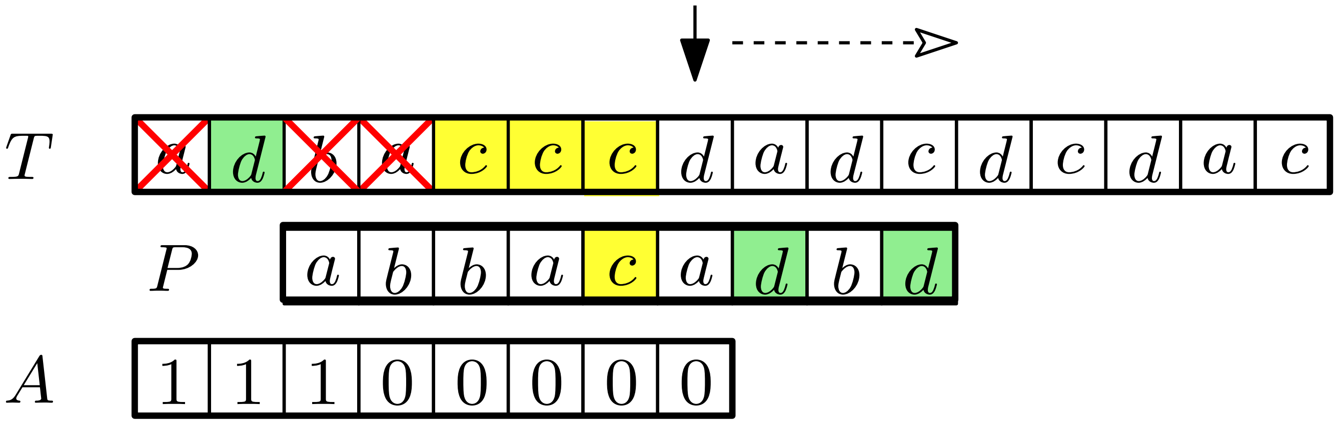
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

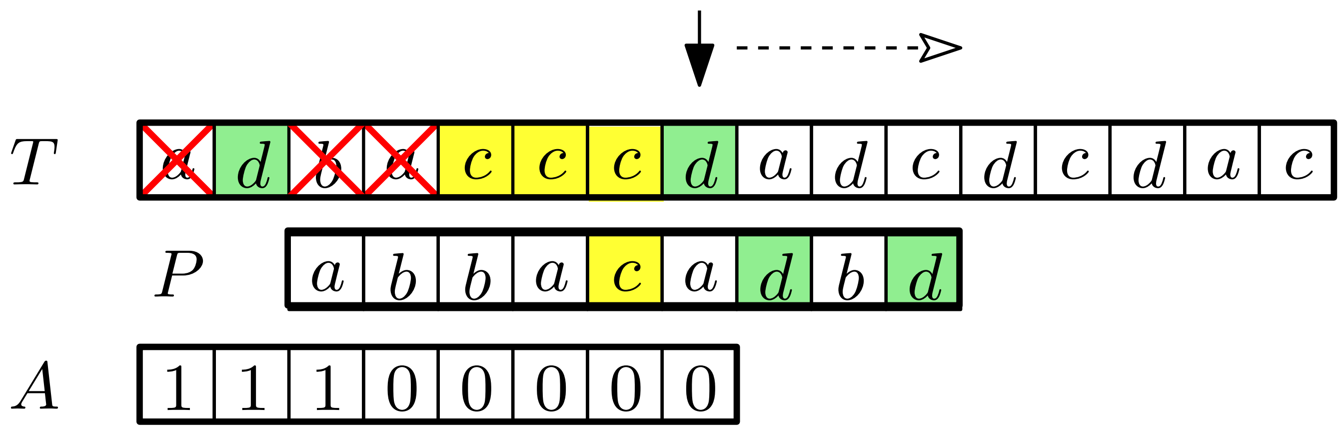
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

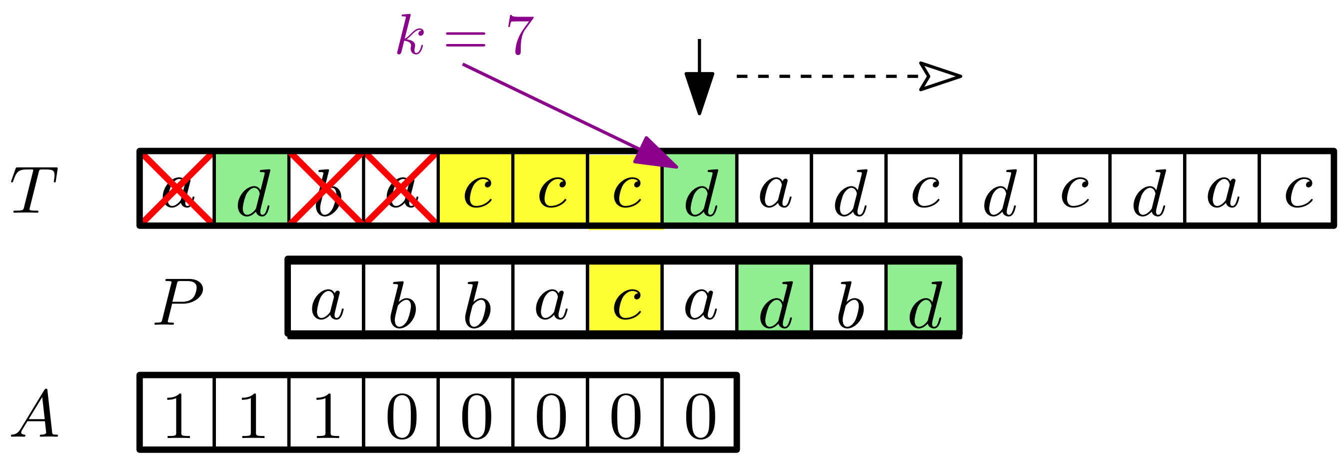
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

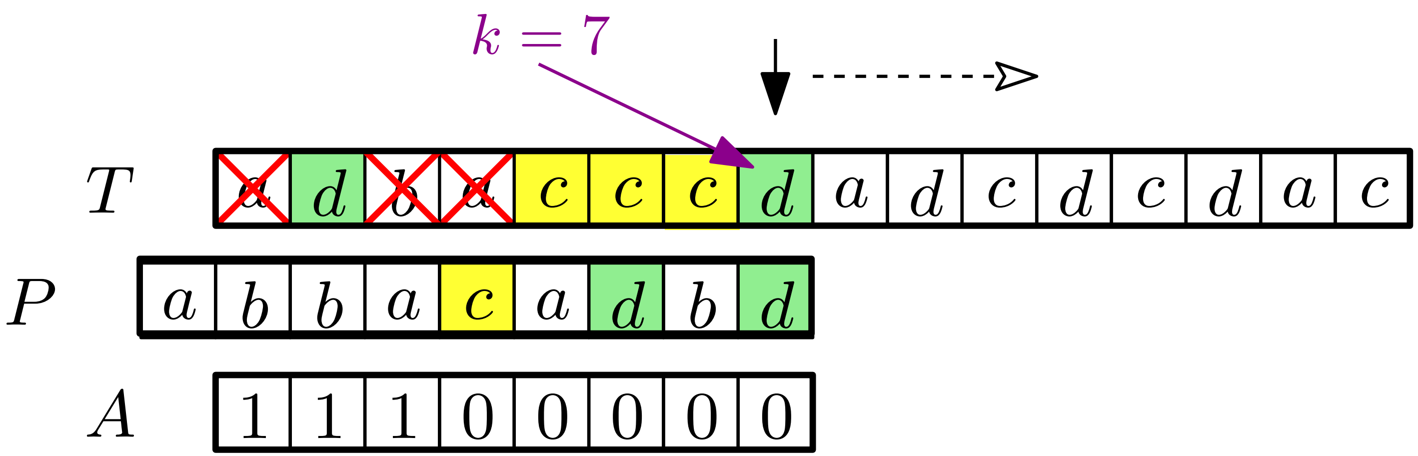
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

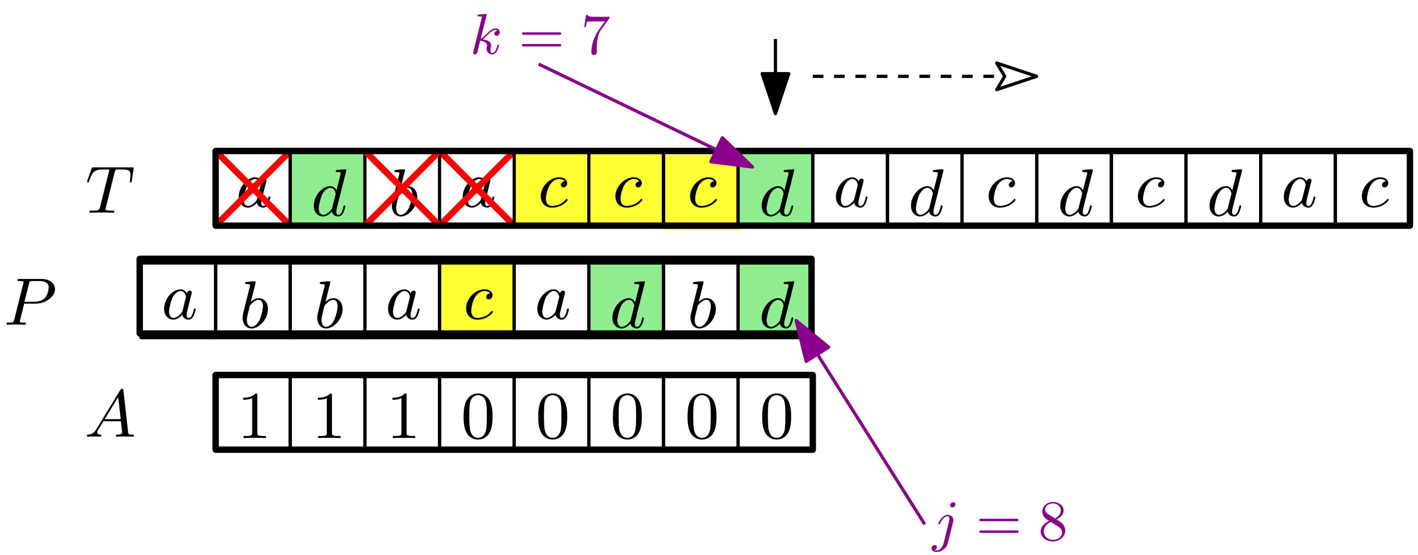
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

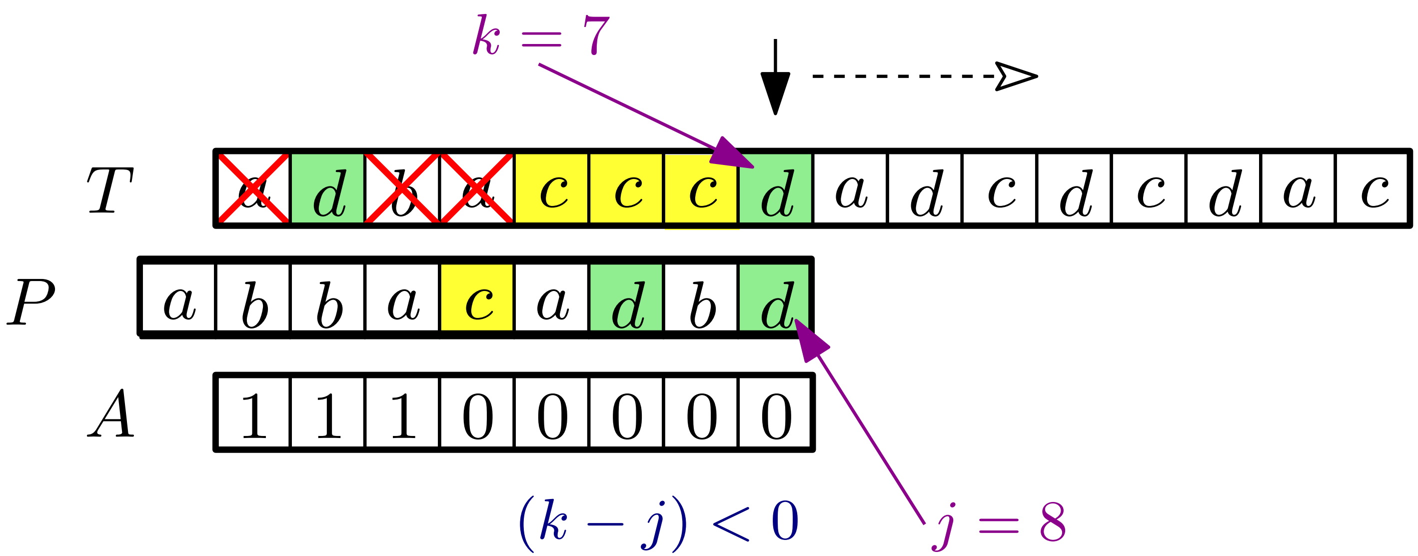
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

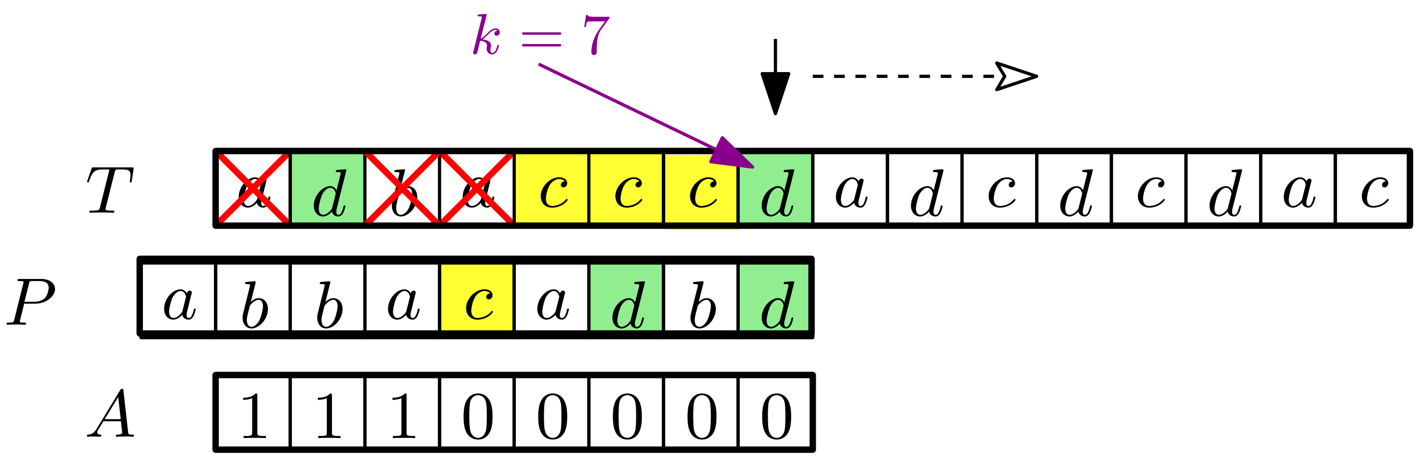
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

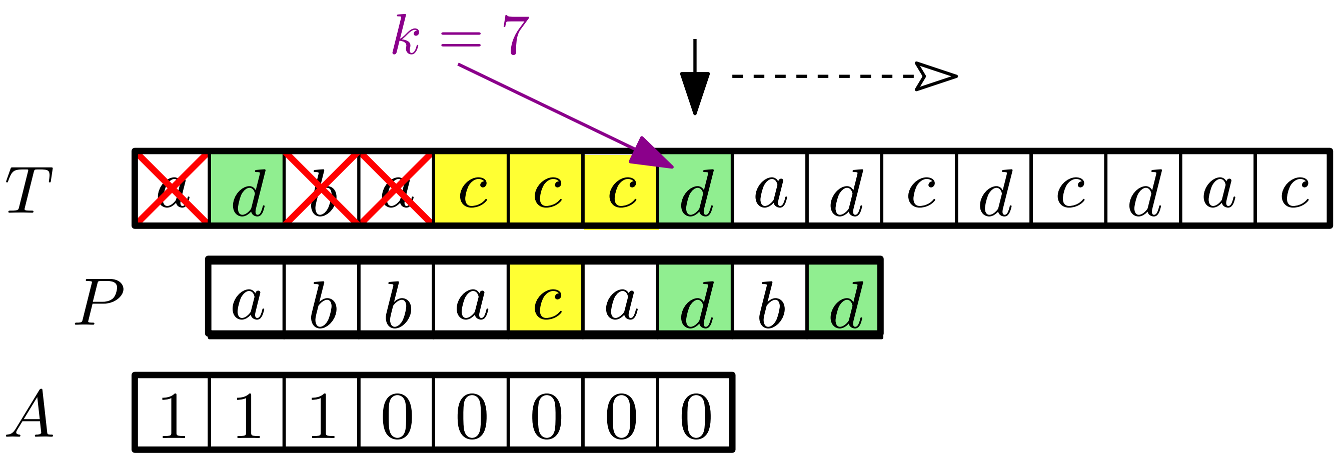
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

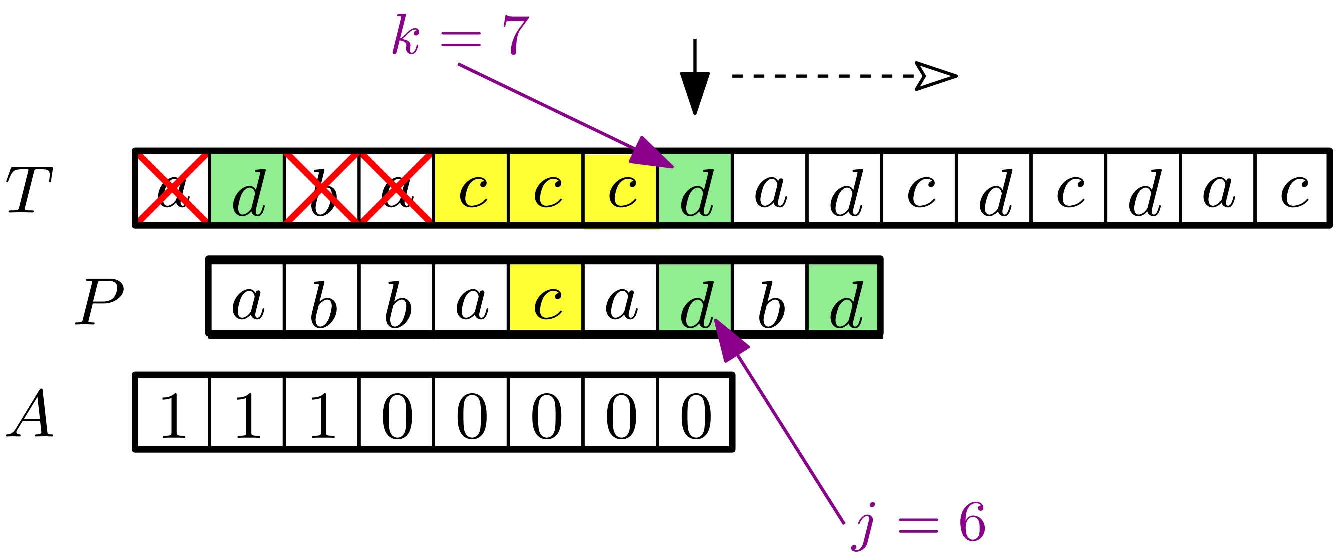
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

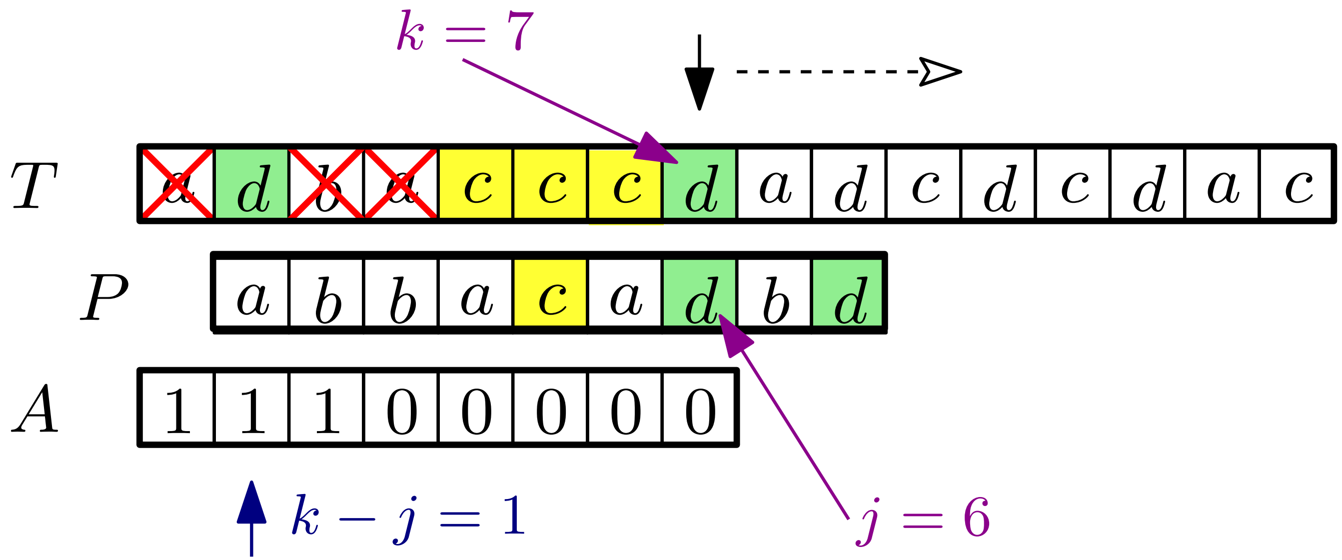
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

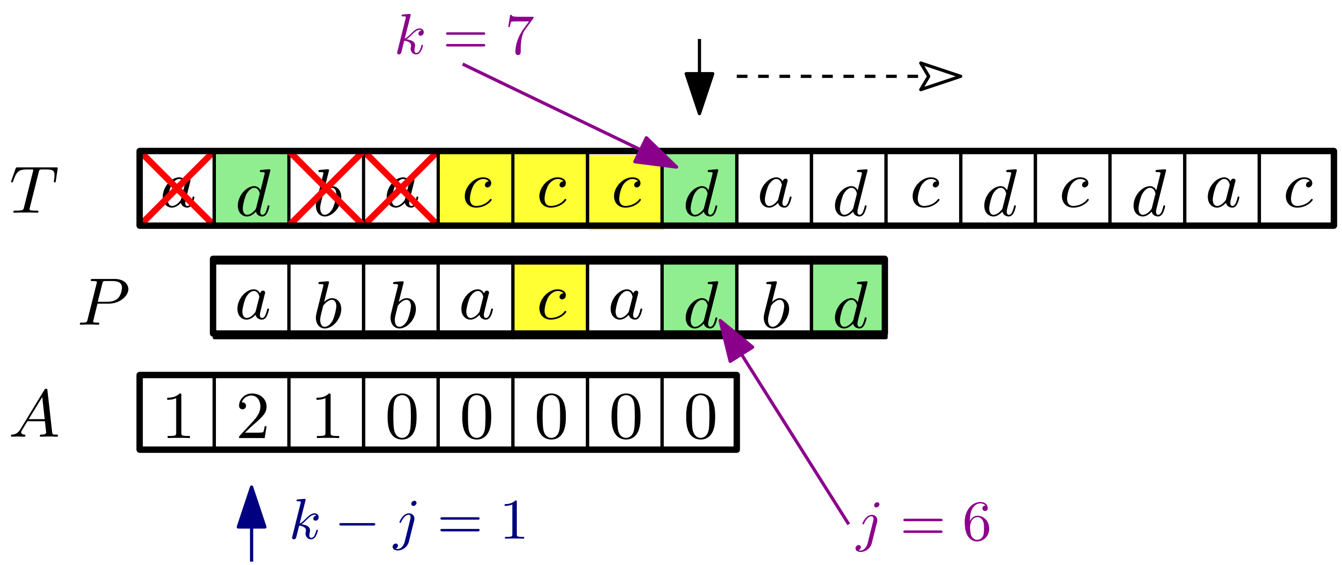
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

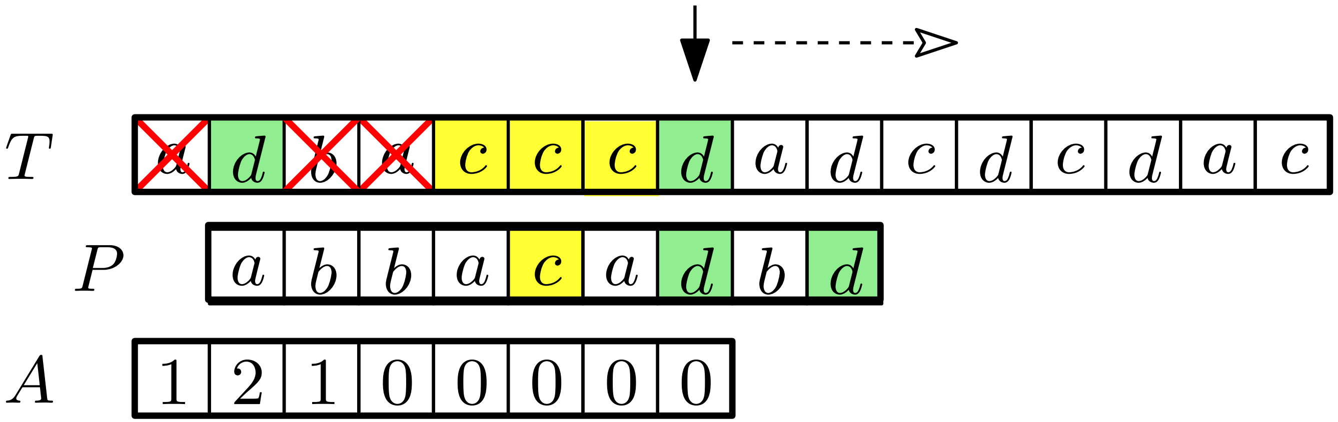
For all j such that $T[k] = P[j]$,

Increase $A[k-j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

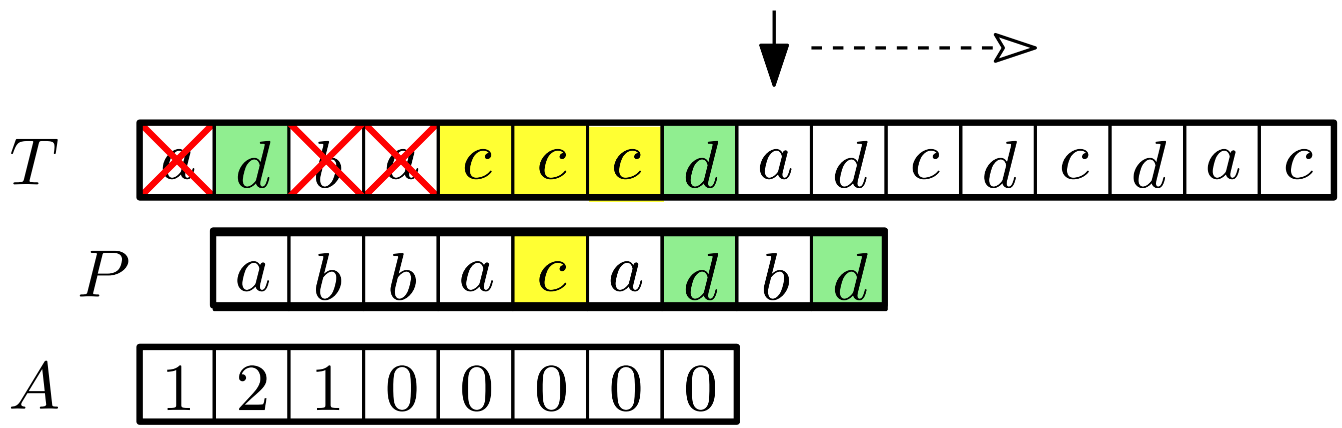
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

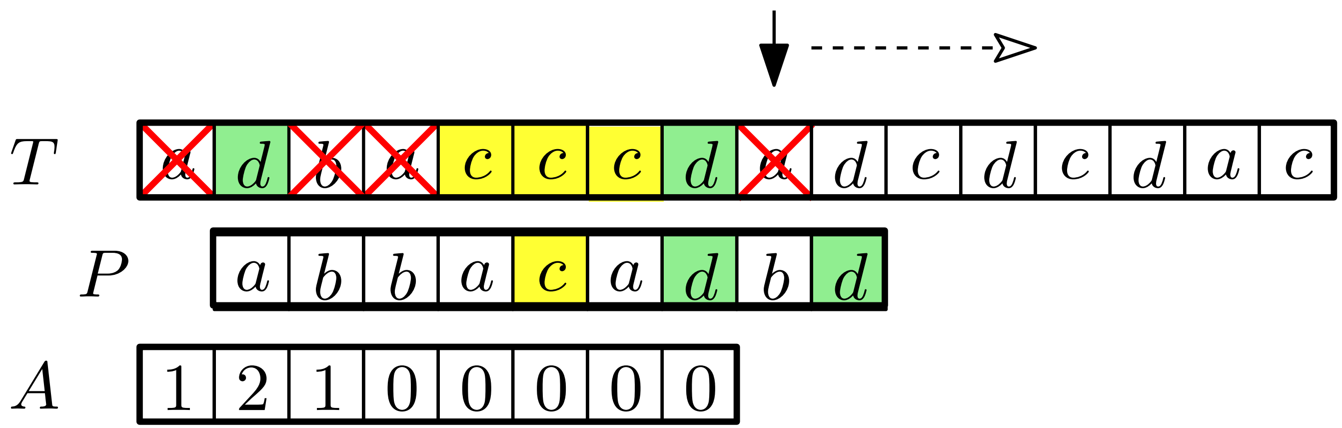
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

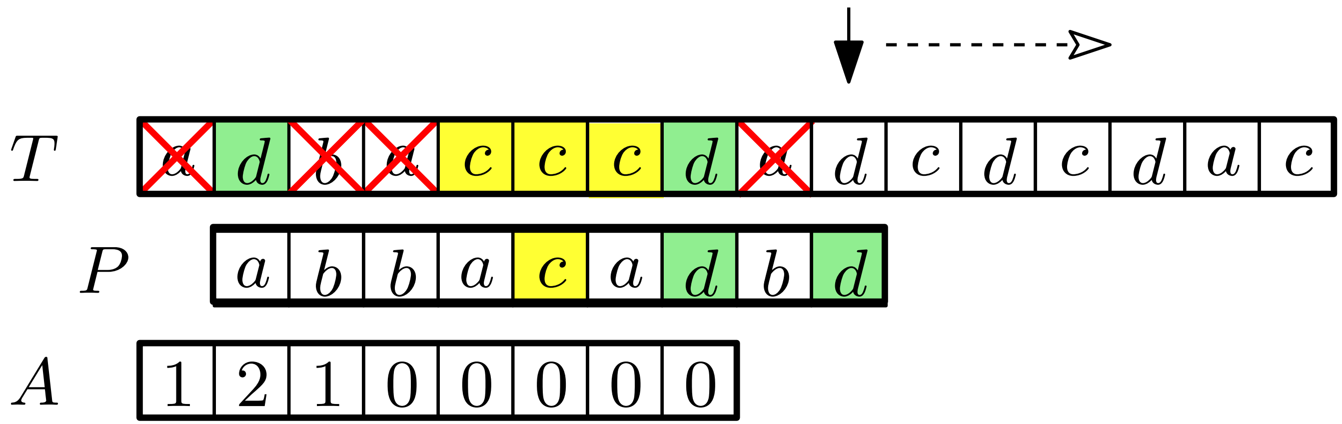
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

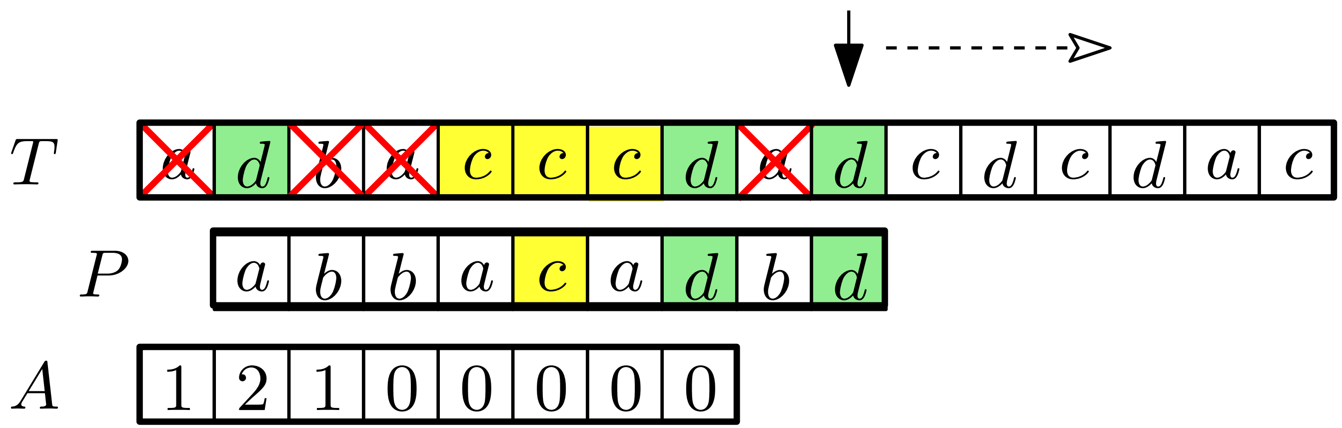
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

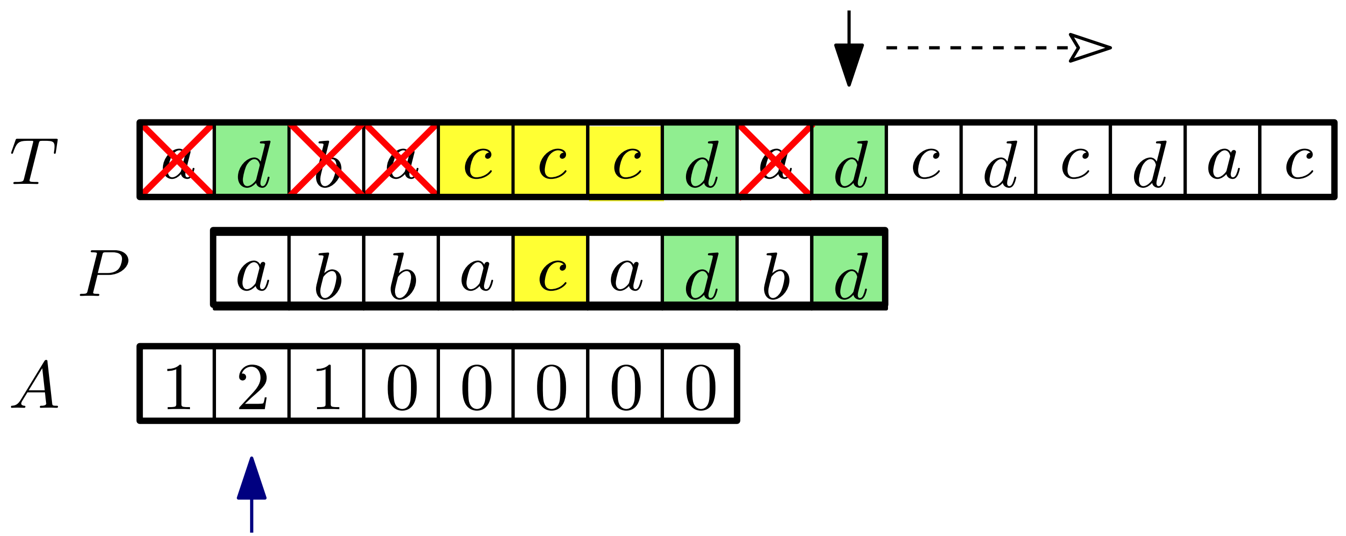
For all j such that $T[k] = P[j]$,

Increase $A[k-j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n-m+1)$ - which is initially all zeros

Make a single pass through $T \ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

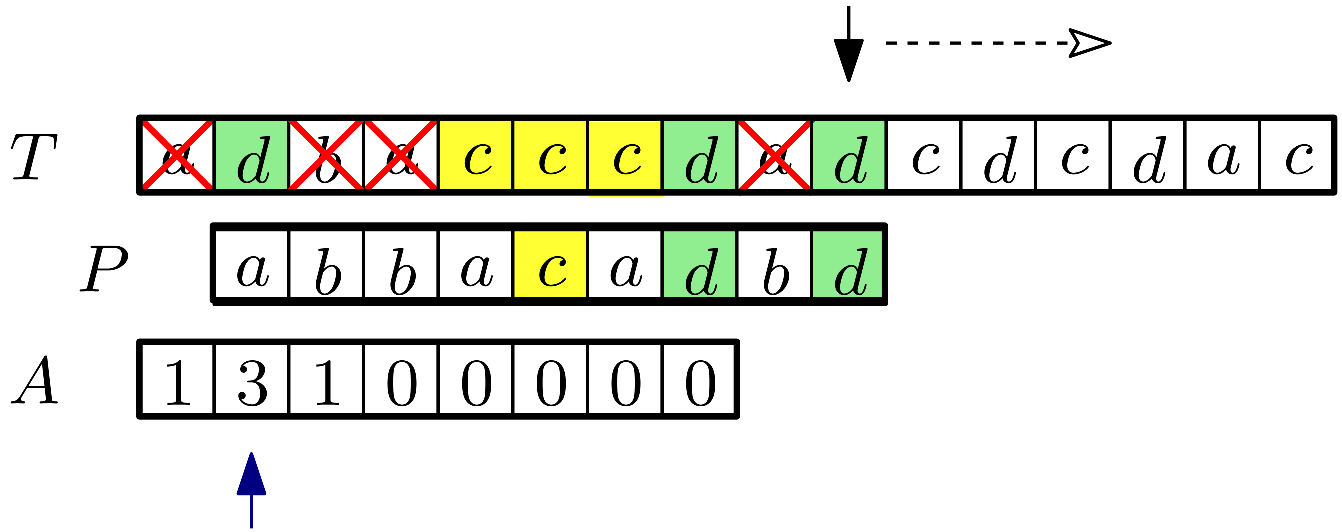
For all j such that $T[k] = P[j]$,

Increase $A[k-j]$ by one

except when $(k-j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

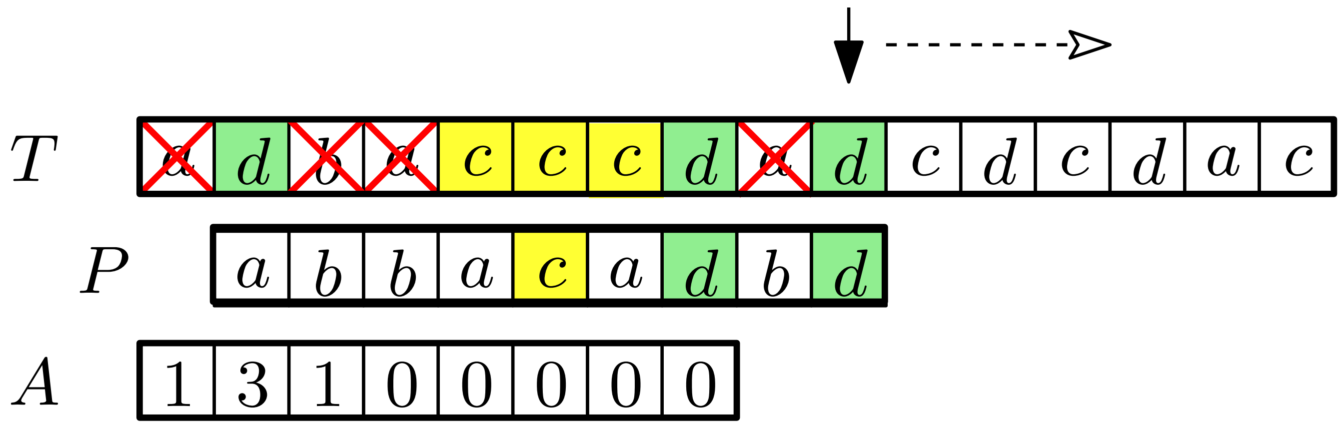
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

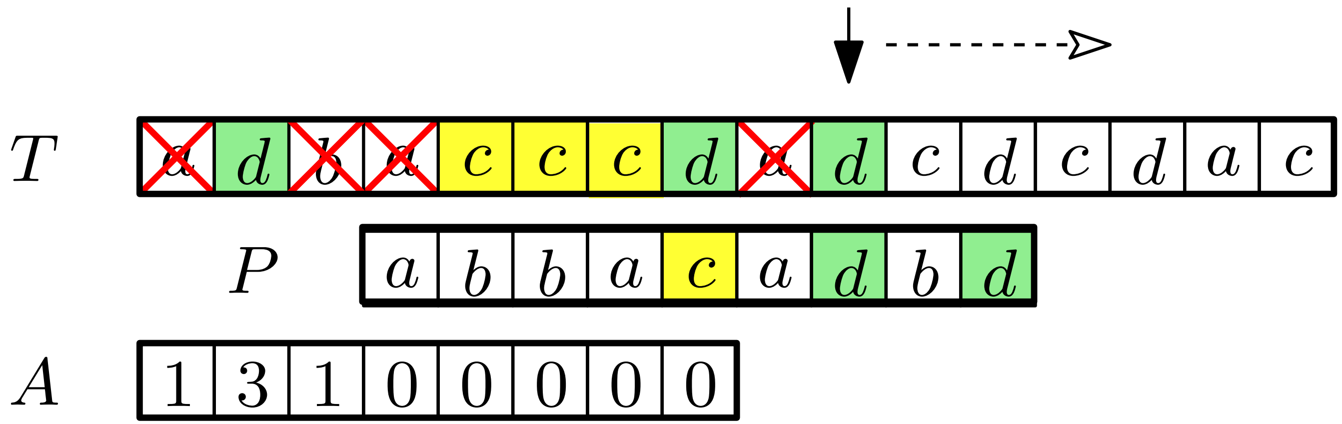
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

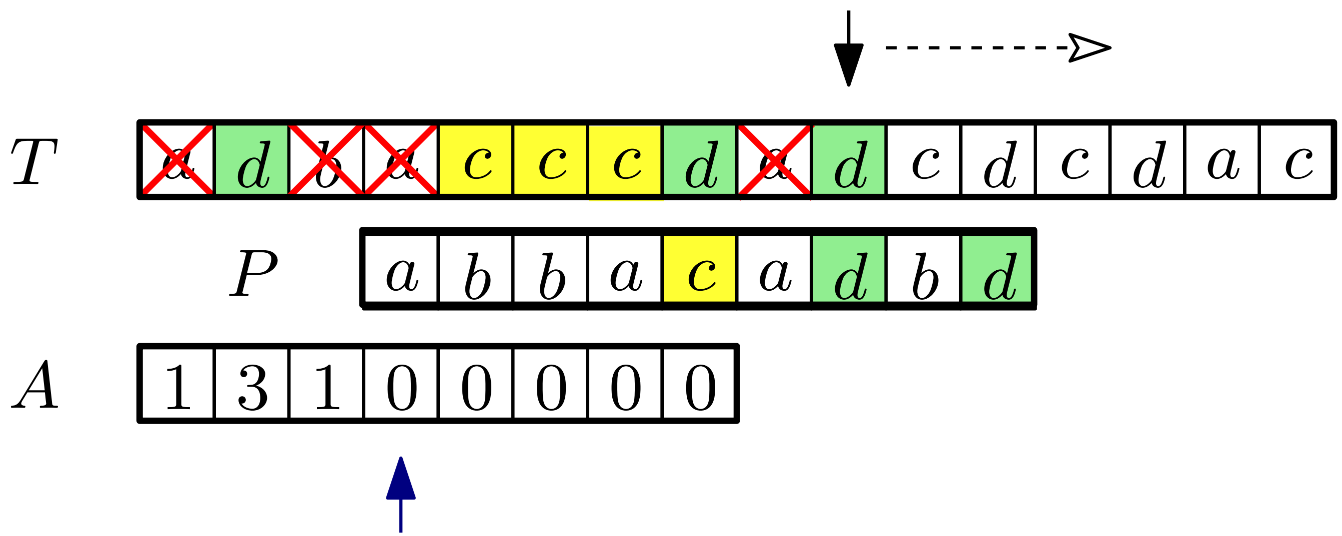
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

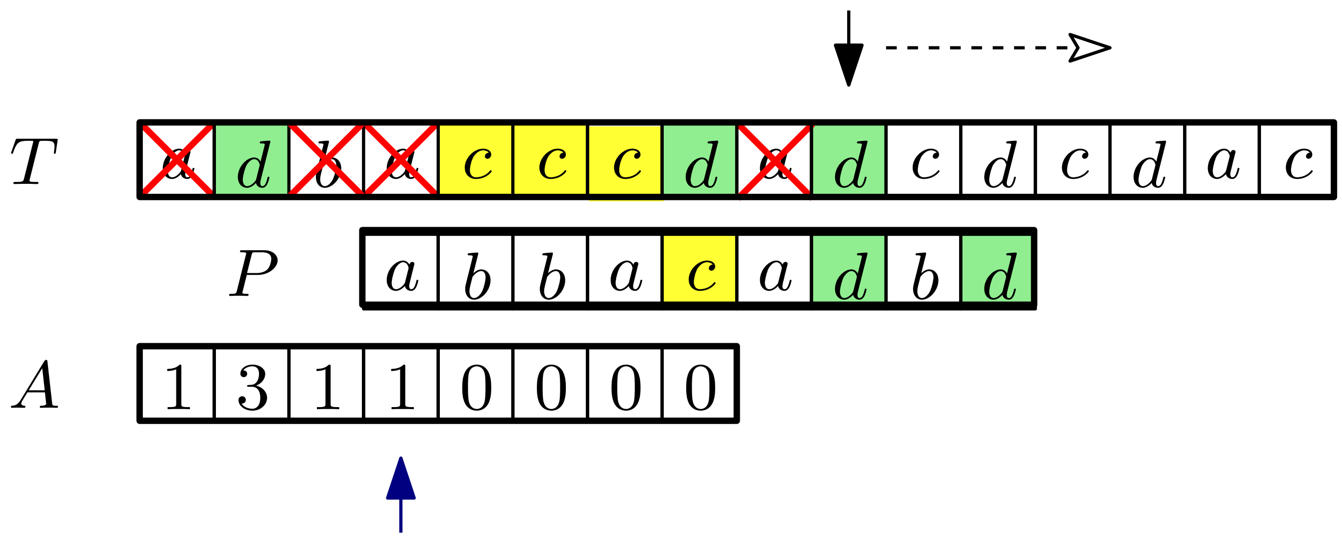
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	~~b~~	~~a~~	c	c	c	d	~~a~~	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	1	3	1	1	0	0	0	0

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

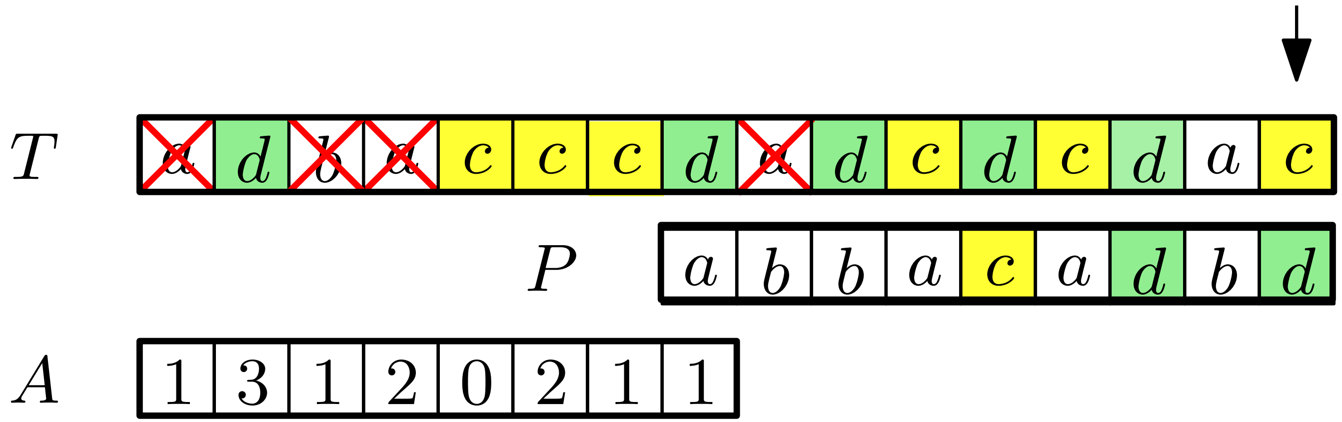
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent. . .

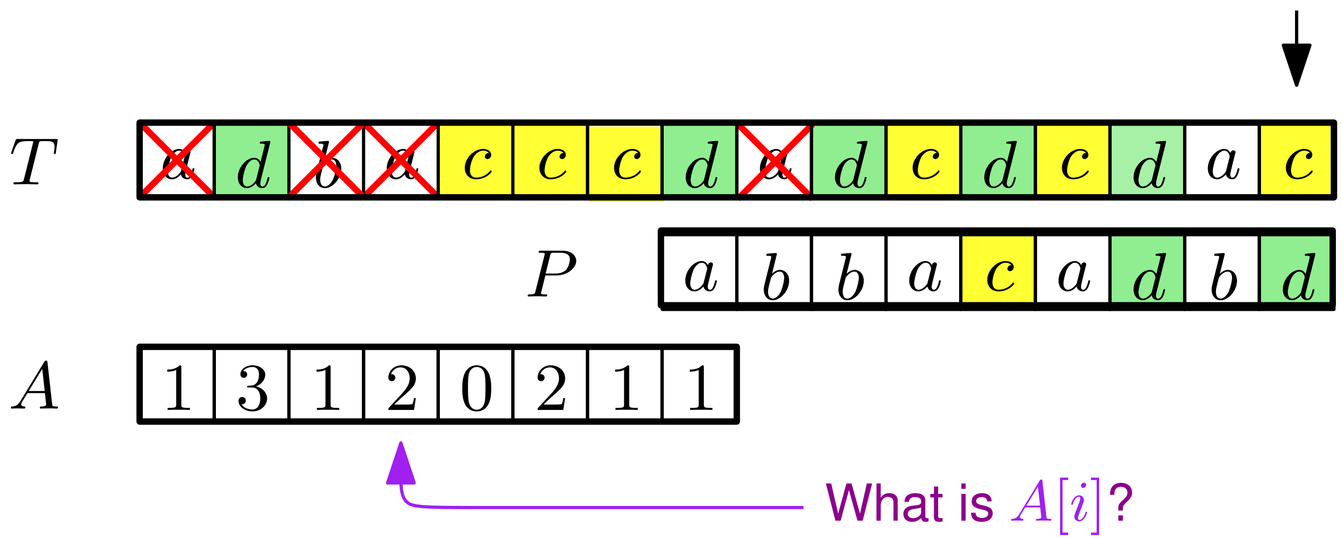
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

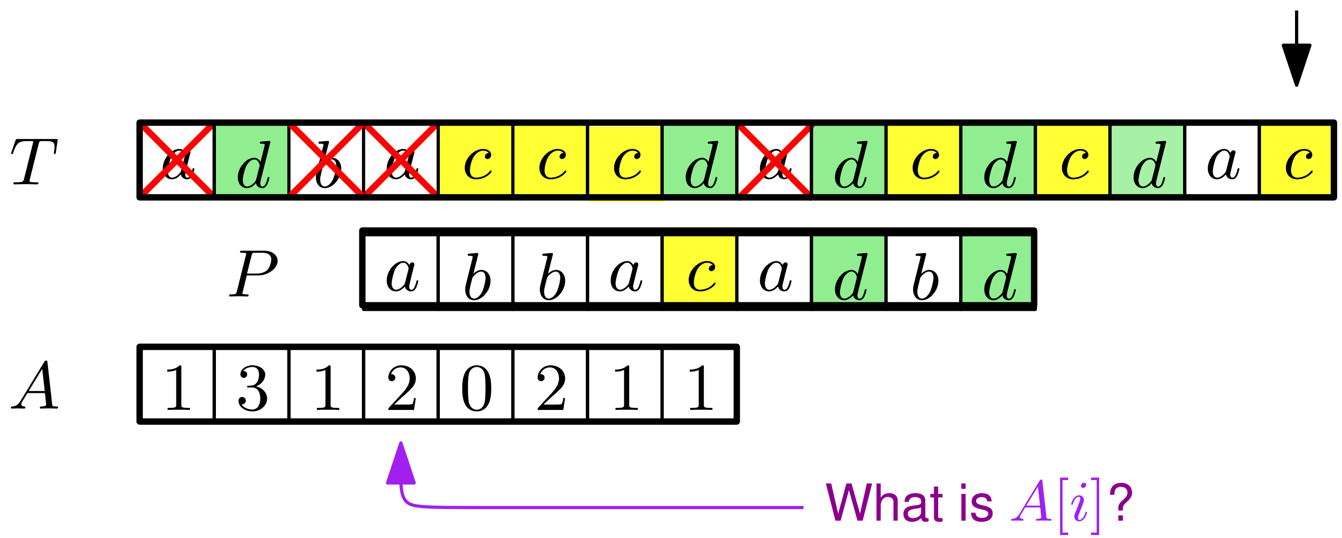
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T\ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

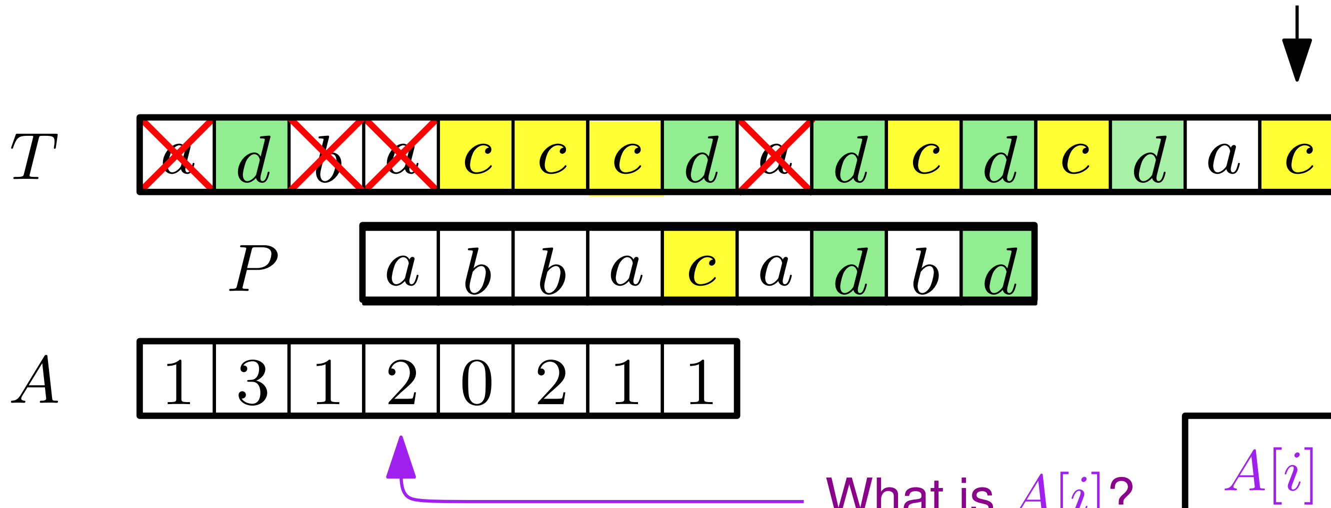
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

$A[i]$ is the number of matches at alignment i involving an infrequent symbol

Step 2: Count all matches involving infrequent symb

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

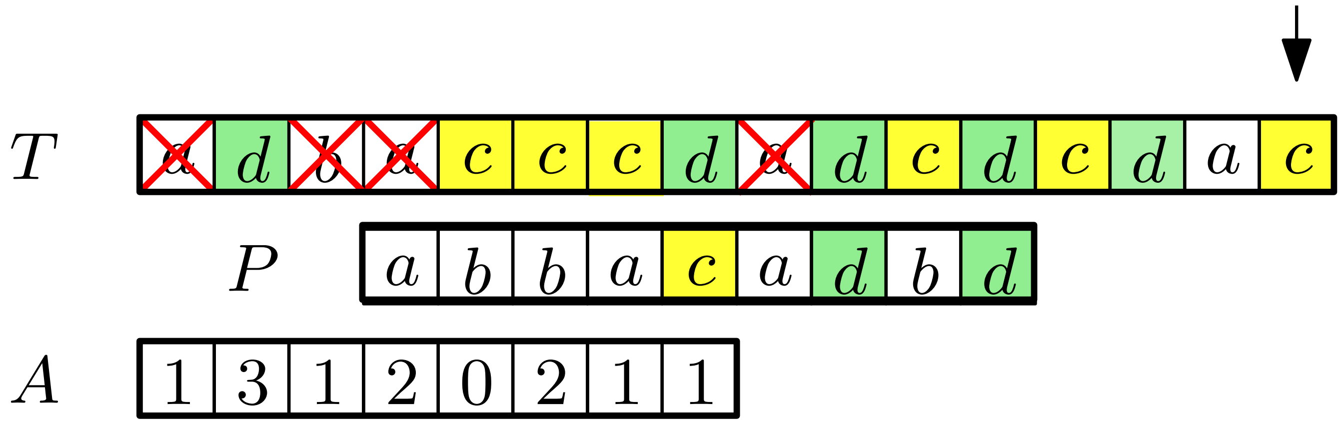
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through $T \ldots$

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

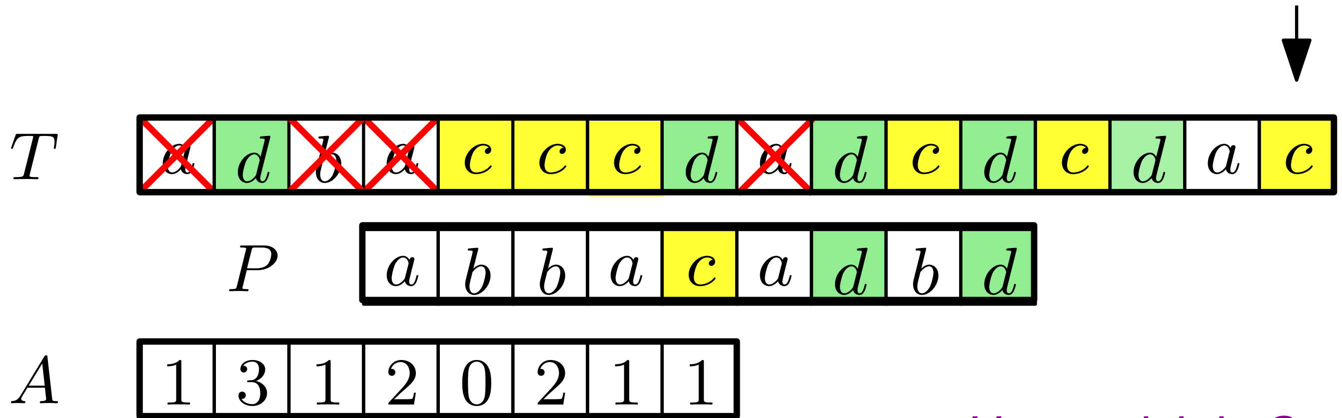
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

How quick is Step 2?

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

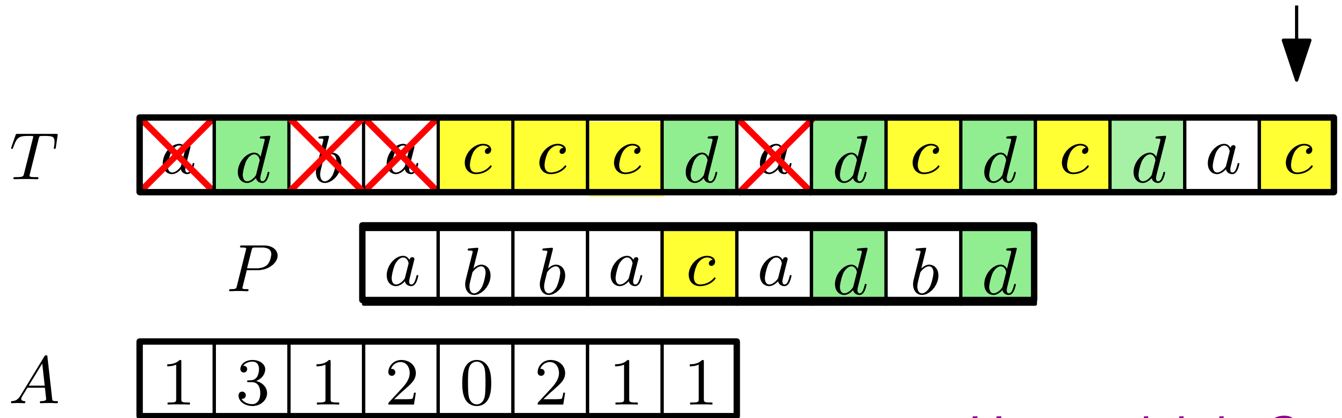
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

How quick is Step 2?

$O(n)$ time

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	~~b~~	~~a~~	c	c	c	d	~~a~~	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	1	3	1	2	0	2	1	1

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

How quick is Step 2?

$O(n)$ time

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

store a list for each infrequent symbol

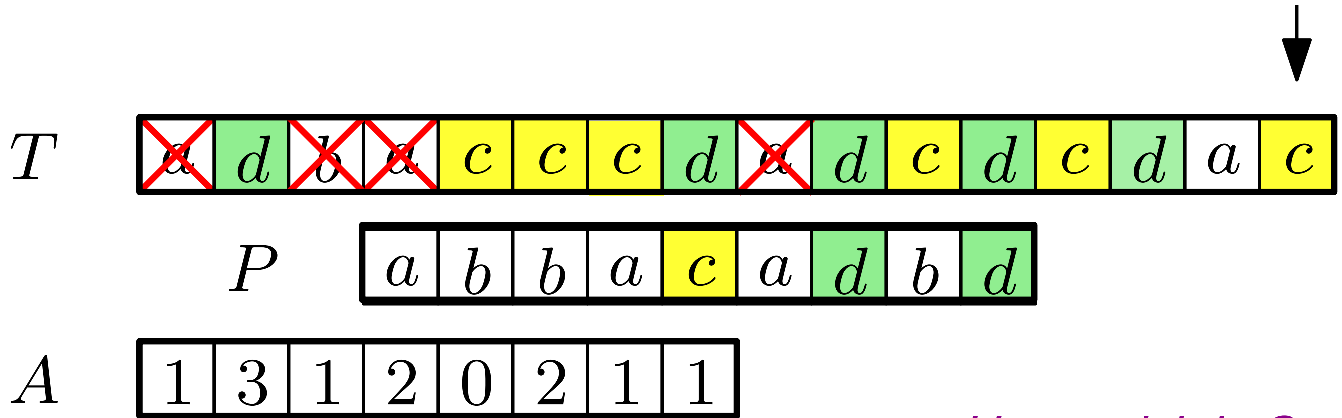
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

How quick is Step 2?

$O(n)$ time

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

store a list for each infrequent symbol

each list has length less than $\sqrt{m}$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	~~a~~	d	~~b~~	~~a~~	c	c	c	d	~~a~~	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	1	3	1	2	0	2	1	1

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

How quick is Step 2?

$O(n)$ time

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

$O(n\sqrt{m})$ time

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

store a list for each infrequent symbol

each list has length less than $\sqrt{m}$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.

T	$\cancel{a}$	d	$\cancel{b}$	$\cancel{a}$	c	c	c	d	$\cancel{a}$	d	c	d	c	d	a	c

P	a	b	b	a	c	a	d	b	d

A	1	3	1	2	0	2	1	1

Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

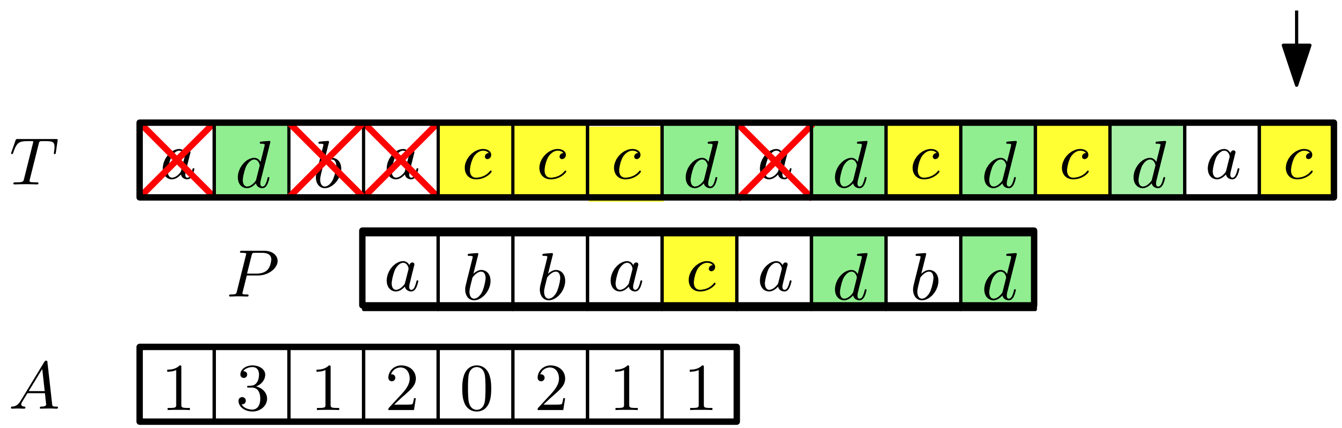
For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

The infrequent/frequent symbols trick

Definition: A symbol is *infrequent* if it occurs fewer than $\sqrt{m}$ times in P.



Every symbol is either frequent or infrequent

a is *frequent*, b is *frequent*
c and d are *infrequent*

Step 2: Count all matches involving infrequent symbols.

Construct an array A of length $(n - m + 1)$ - which is initially all zeros

Make a single pass through T...

For each character $T[k]$, (where $0 \leqslant k < n$)

If $T[k]$ is infrequent...

For all j such that $T[k] = P[j]$,

Increase $A[k - j]$ by one

except when $(k - j) < 0$

$O(n\sqrt{m})$ total time

Pattern matching with mismatches: putting it all together

Algorithm summary

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Step 1: Count all matches involving frequent symbols. ($O(n\sqrt{m} \log m)$ time)

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Step 1: Count all matches involving frequent symbols. ($O(n\sqrt{m} \log m)$ time)

Step 2: Count all matches involving infrequent symbols. ($O(n\sqrt{m})$ time)

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Step 1: Count all matches involving frequent symbols. ($O(n\sqrt{m} \log m)$ time)

Step 2: Count all matches involving infrequent symbols. ($O(n\sqrt{m})$ time)

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Step 1: Count all matches involving frequent symbols. ($O(n\sqrt{m} \log m)$ time)

Step 2: Count all matches involving infrequent symbols. ($O(n\sqrt{m})$ time)

at any alignment

the number of mismatches is just m minus the total number of matches

Pattern matching with mismatches: putting it all together

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent ($O(m \log m)$ time)

Step 1: Count all matches involving frequent symbols. ($O(n\sqrt{m} \log m)$ time)

Step 2: Count all matches involving infrequent symbols. ($O(n\sqrt{m})$ time)

at any alignment
the number of mismatches is just m minus the total number of matches

Overall, we obtain a time complexity of $O(n\sqrt{m} \log m)$.