Advanced Algorithms – COMS31900

2013/2014

Lecture 13

Approximate pattern matching (part two)

Benjamin Sach

University of Bristol

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$m = 9$

$k = 4$

$(\sqrt{k} = 2)$

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$
$(\sqrt{k} = 2)$

a is *frequent*

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

a is *frequent*, b is *frequent*

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

a is *frequent*, b is *frequent*, d is *frequent*

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$$k = 4$$
$$(\sqrt{k} = 2)$$

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

How many frequent symbols can there be?

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots!

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

a is *frequent*, b is *frequent*, d is *frequent*

c is *infrequent*

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$
$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

Algorithm summary

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

Algorithm summary

Step 0: Classify each symbol as frequent or infrequent

Step 1: Count all matches involving frequent symbols (using convolutions)

Step 2: Count all matches involving infrequent symbols (as before)

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

Algorithm summary

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

Step 1: Count all matches involving frequent symbols (using convolutions)

Step 2: Count all matches involving infrequent symbols (as before)

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P,
and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

Algorithm summary

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

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

Step 2: Count all matches involving infrequent symbols (as before)

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

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

$k = 4$

$(\sqrt{k} = 2)$

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

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P.

Algorithm summary

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

Step 1: Count all matches involving frequent symbols (using convolutions)

- $O(n\sqrt{k} \log m)$ time

Step 2: Count all matches involving infrequent symbols (as before)

- $O(n\sqrt{k})$ time

k-mismatch using frequent/infrequent symbols

Definition: A symbol is *frequent* if it occurs at least $\sqrt{k}$ times in P, and infrequent otherwise

P

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

$k = 4$

$(\sqrt{k} = 2)$

a is *frequent*, b is *frequent*, d is *frequent*

c is *infrequent*

How many frequent symbols can there be? Lots! there could be $\frac{m}{\sqrt{k}}$ frequent symbols

---

Case 1: There are fewer than $2\sqrt{k}$ frequent symbols in P. - $O(n\sqrt{k}\log m)$ total time

Algorithm summary

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

Step 1: Count all matches involving frequent symbols (using convolutions)

- $O(n\sqrt{k}\log m)$ time

Step 2: Count all matches involving infrequent symbols (as before)

- $O(n\sqrt{k})$ time

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

P	a	e	b	b	a	c	a	d	b	d	c	f	b	b

$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

P	a	e	b	b	a	c	a	d	b	d	c	f	b	b
	0	1	2	3	4	5	6	7	8	9	10	11	12	13

$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

P	a	e	b	b	a	c	a	d	b	d	c	f	b	b
	0	1	2	3	4	5	6	7	8	9	10	11	12	13

$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

P	a	e	b	b	a	c	a	d	b	d	c	f	b	b
	0	1	2	3	4	5	6	7	8	9	10	11	12	13

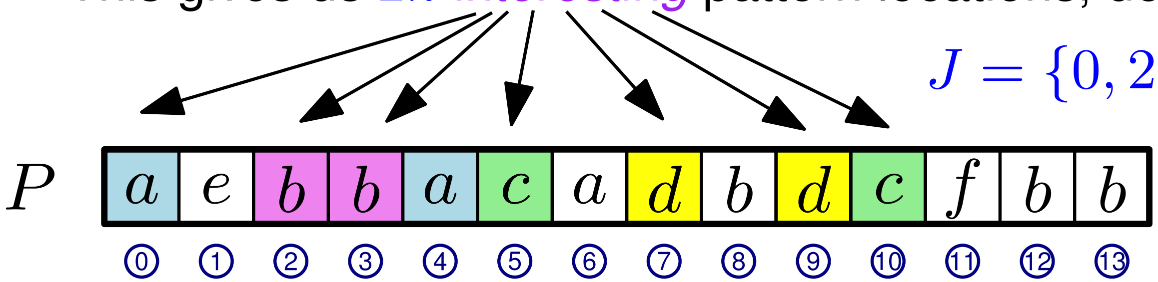
$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

$J = \{0, 2, 3, 4, 5, 7, 9, 10\}$



$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

$J = \{0, 2, 3, 4, 5, 7, 9, 10\}$

P														
	a	e	b	b	a	c	a	d	b	d	c	f	b	b

$k = 4$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J

$J = \{0, 2, 3, 4, 5, 7, 9, 10\}$

P	a	e	b	b	a	c	a	d	b	d	c	f	b	b

$k = 4$

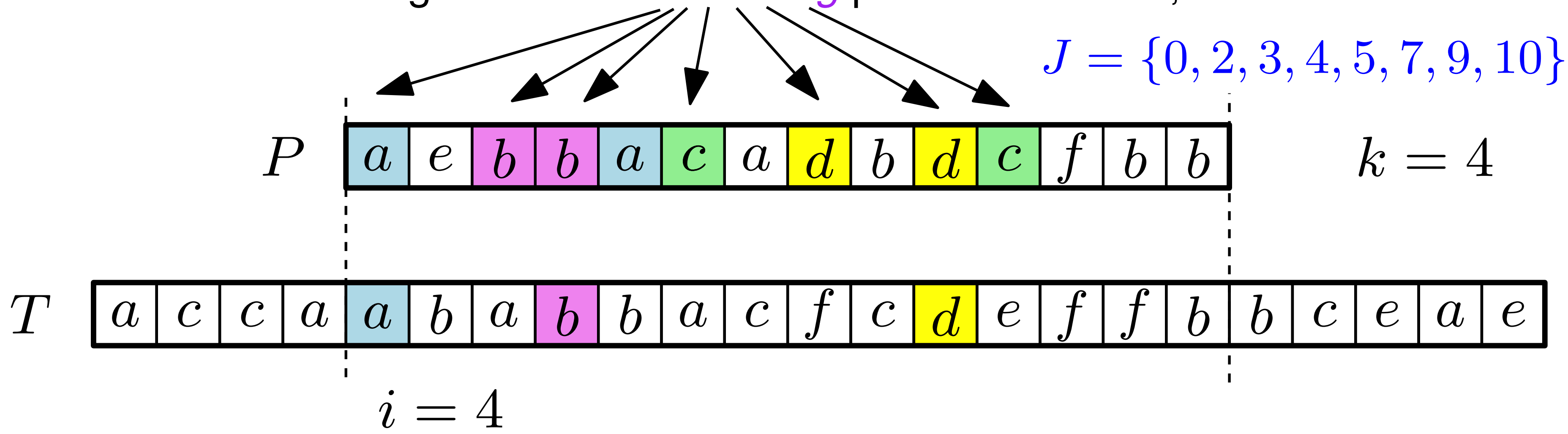
Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



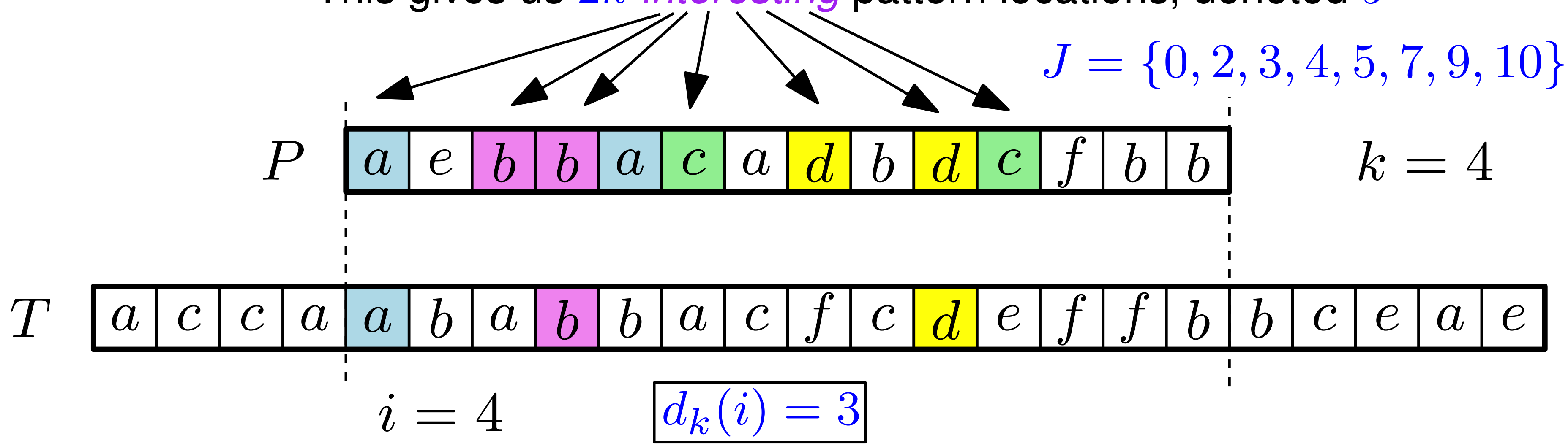
Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



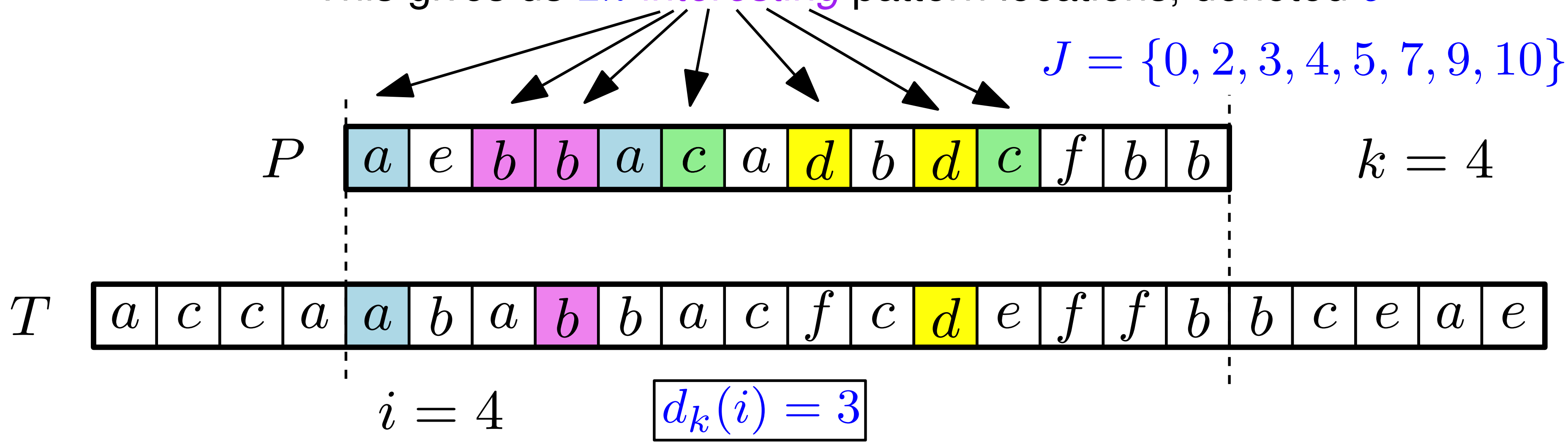
Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

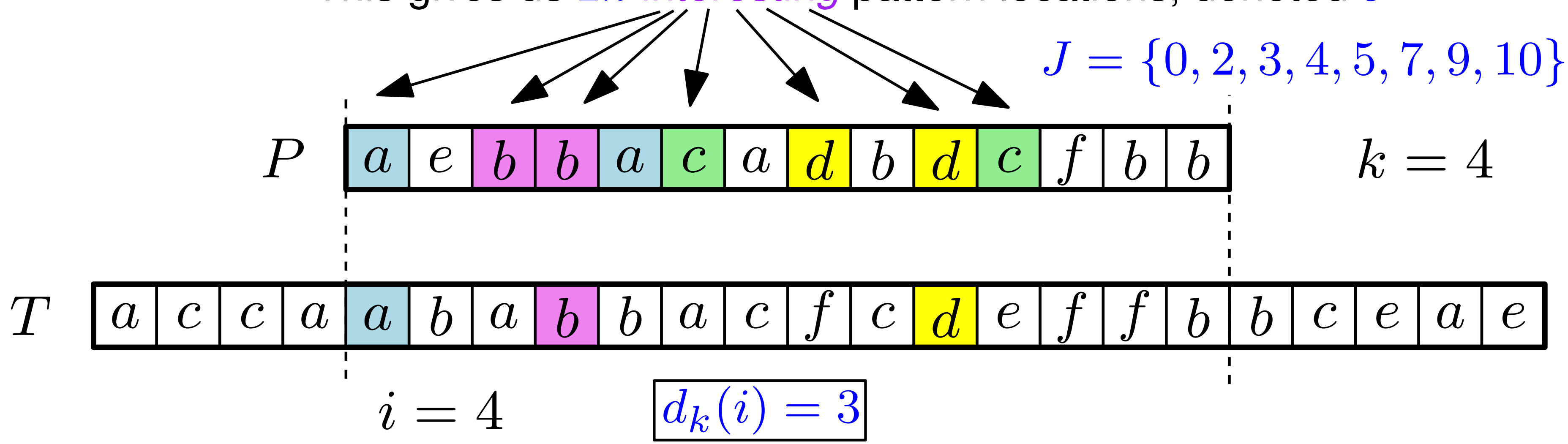
i.e. the number of (single character) matches involving interesting pattern locations

Fact if $d_k(i) < k$ then there are more than k mismatches (i.e. $\text{Ham}_k(i) = X$)

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i+j]$

i.e. the number of (single character) matches involving interesting pattern locations

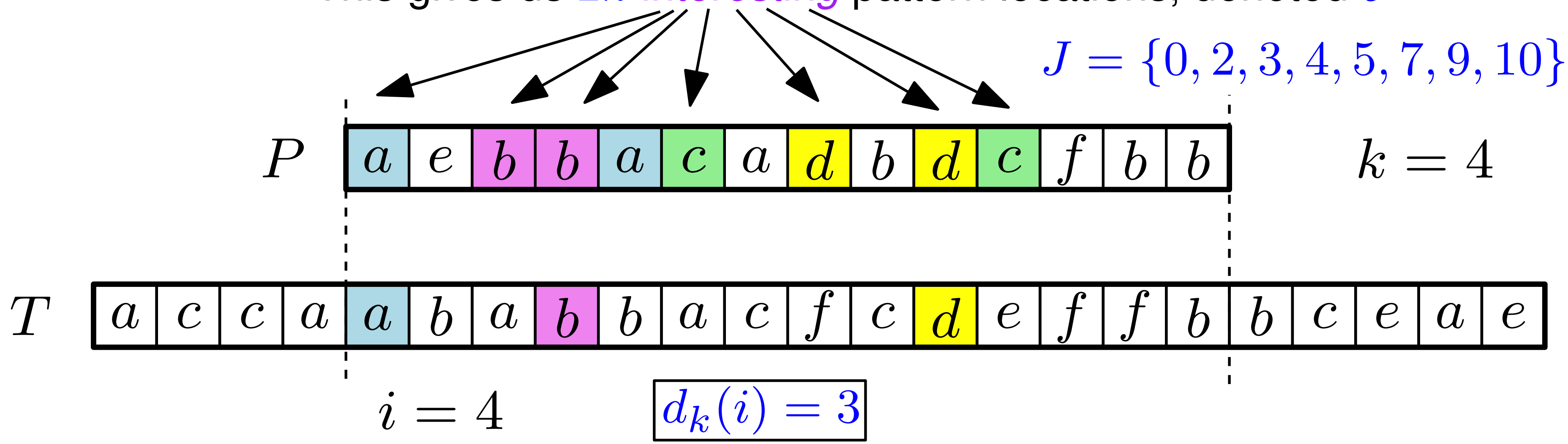
Fact if $d_k(i) < k$ then there are more than k mismatches (i.e. $\text{Ham}_k(i) = X$)

because there are $2k$ *interesting positions... and fewer than* k *of them match*

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact if $d_k(i) < k$ then there are more than k mismatches (i.e. $\text{Ham}_k(i) = X$)

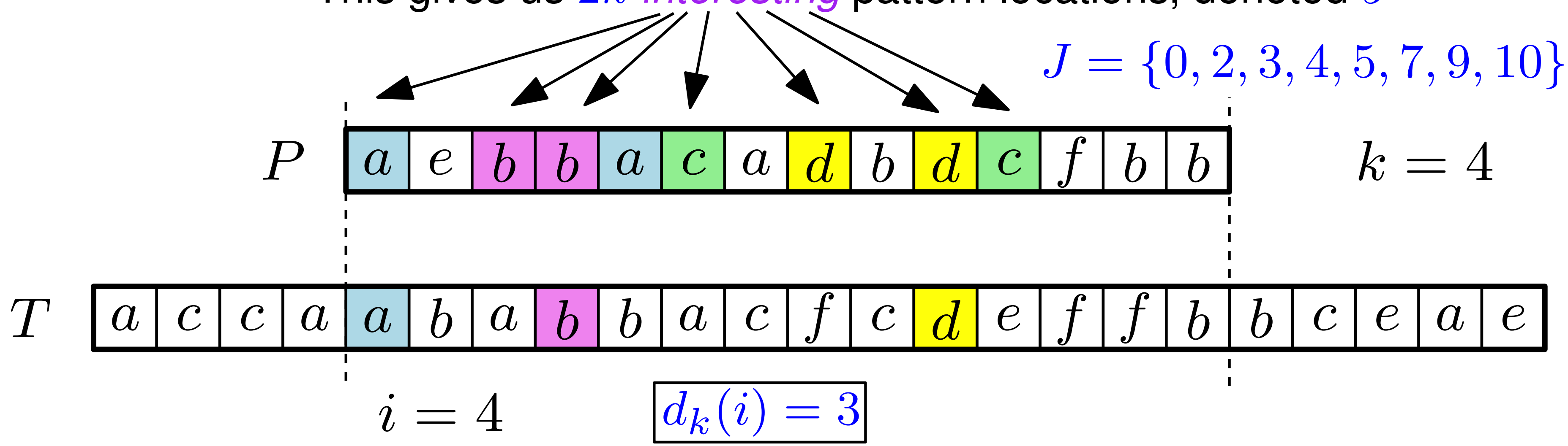
because there are $2k$ interesting positions... and fewer than k of them match

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact if $d_k(i) < k$ then there are more than k mismatches (i.e. $\text{Ham}_k(i) = X$)

because there are $2k$ interesting positions. . . and fewer than k of them match

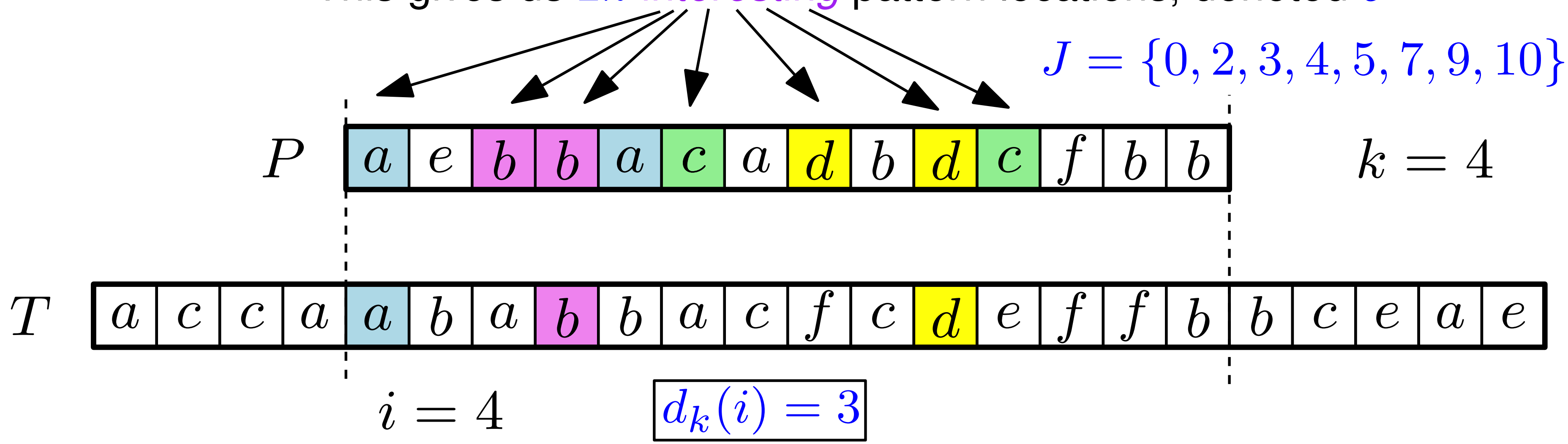
Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

this follows from a counting argument

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

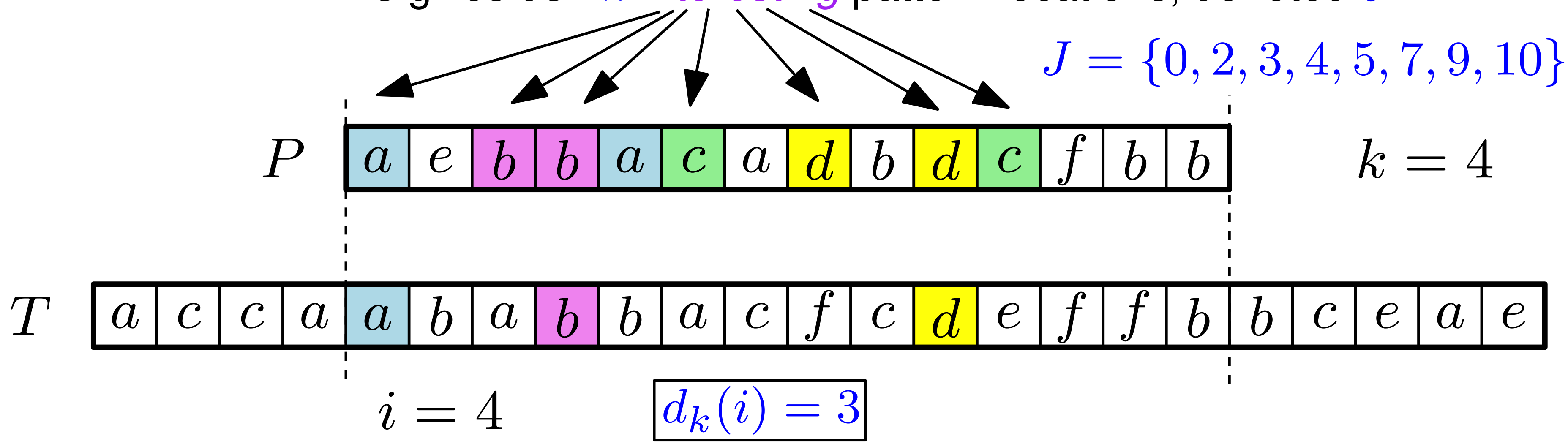
i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

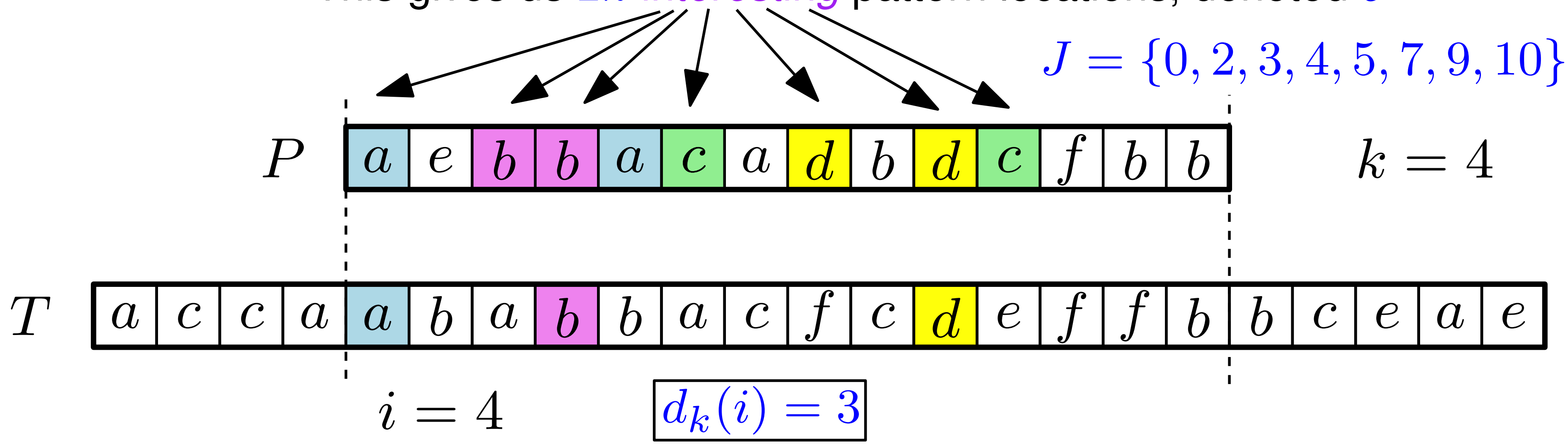
Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

For any location i', $T[i'] = P[j]$ for either 0 or $\sqrt{k}$ distinct $j \in J$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i+j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

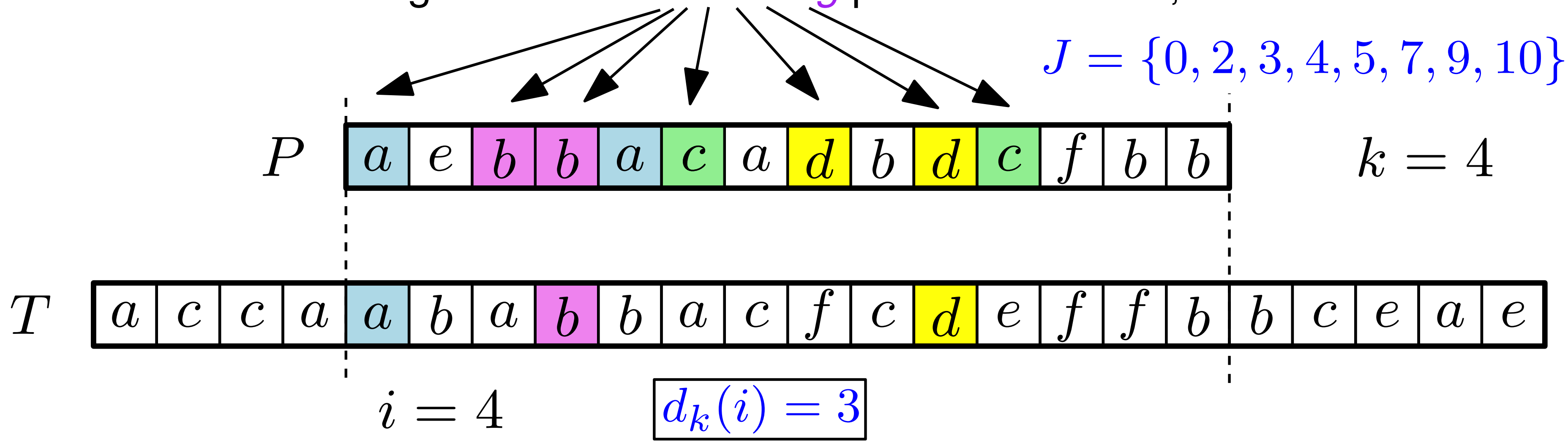
For any location i', $T[i'] = P[j]$ for either 0 or $\sqrt{k}$ distinct $j \in J$

This implies that $\sum_i d_k(i) \leqslant \sum_{i'} \sum_{j \in J} \text{Eq}(T[i'], P[j]) \leqslant n\sqrt{k}$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interes

Eq $= 1$ if $T[i'] = P[j]$
(and 0 otherwise)

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

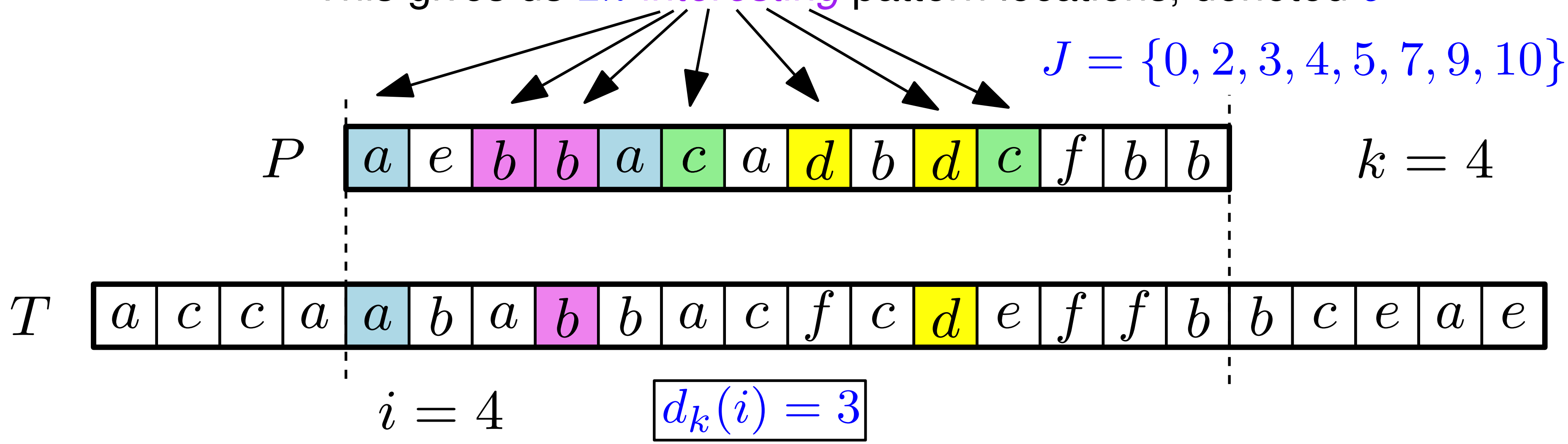
For any location i', $T[i'] = P[j]$ for either 0 or $\sqrt{k}$ distinct $j \in J$

This implies that $\sum_i d_k(i) \leqslant \sum_{i'} \sum_{j \in J}$ Eq$(T[i'], P[j]) \leqslant n\sqrt{k}$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i+j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

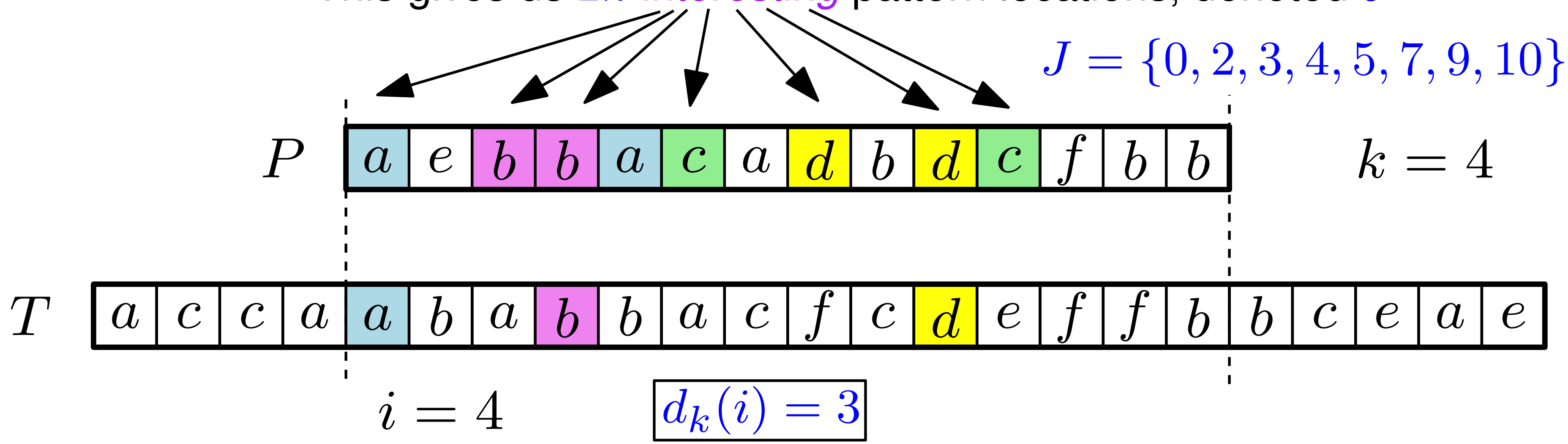
For any location i', $T[i'] = P[j]$ for either 0 or $\sqrt{k}$ distinct $j \in J$

This implies that $\sum_i d_k(i) \leqslant \sum_{i'} \sum_{j \in J} \mathsf{Eq}(T[i'], P[j]) \leqslant n\sqrt{k}$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

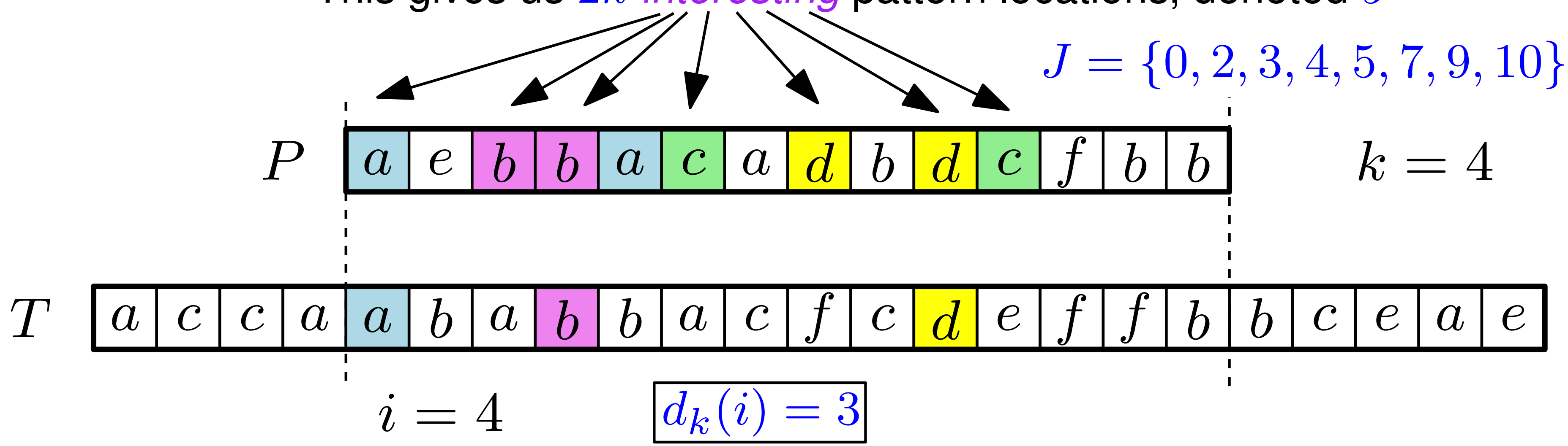
Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

Assume that more than $n/\sqrt{k}$ values of i have $d_k(i) \geqslant k$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

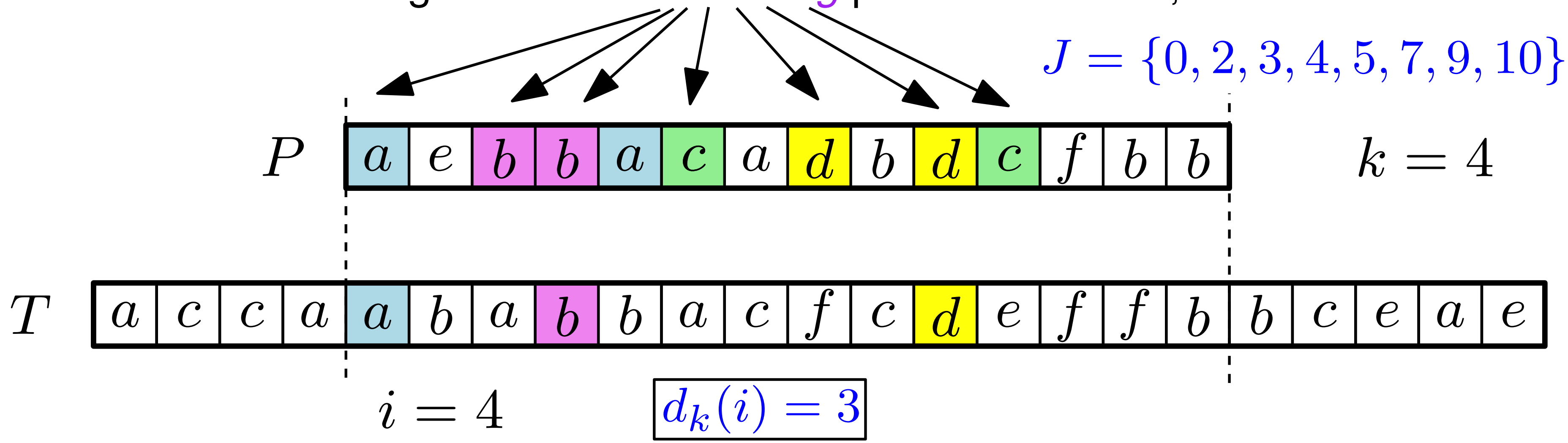
Assume that more than $n/\sqrt{k}$ values of i have $d_k(i) \geqslant k$

So $\sum_i d_k(i) \geqslant \frac{n}{\sqrt{k}} \cdot k$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i+j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

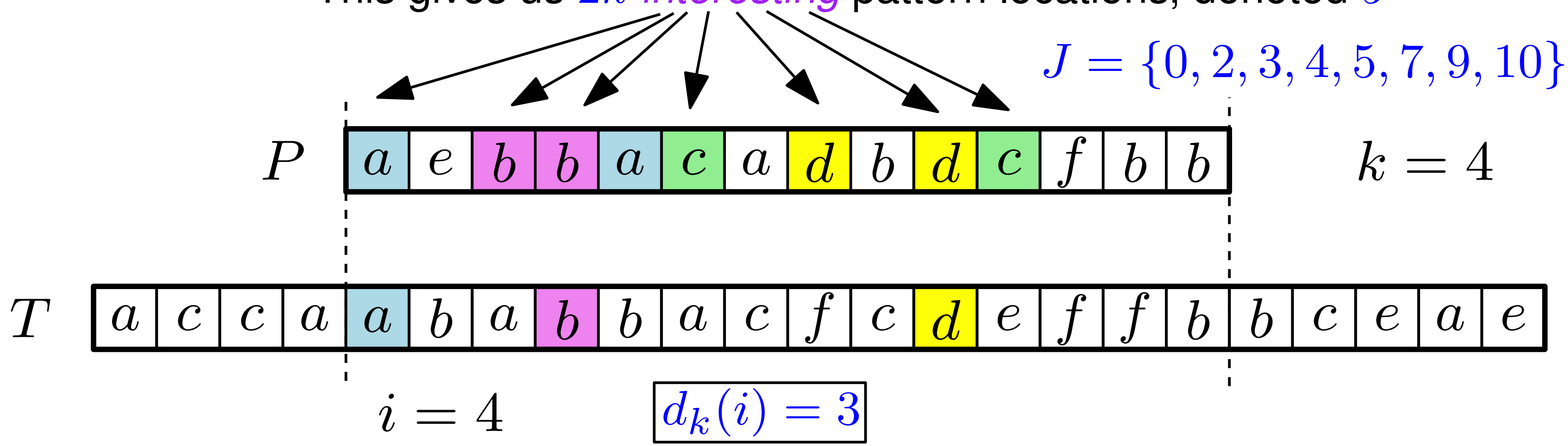
Assume that more than $n/\sqrt{k}$ values of i have $d_k(i) \geqslant k$

So $\sum_i d_k(i) \geqslant \frac{n}{\sqrt{k}} \cdot k > n\sqrt{k}$

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$

i.e. the number of (single character) matches involving interesting pattern locations

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$

Assume that more than $n/\sqrt{k}$ values of i have $d_k(i) \geqslant k$

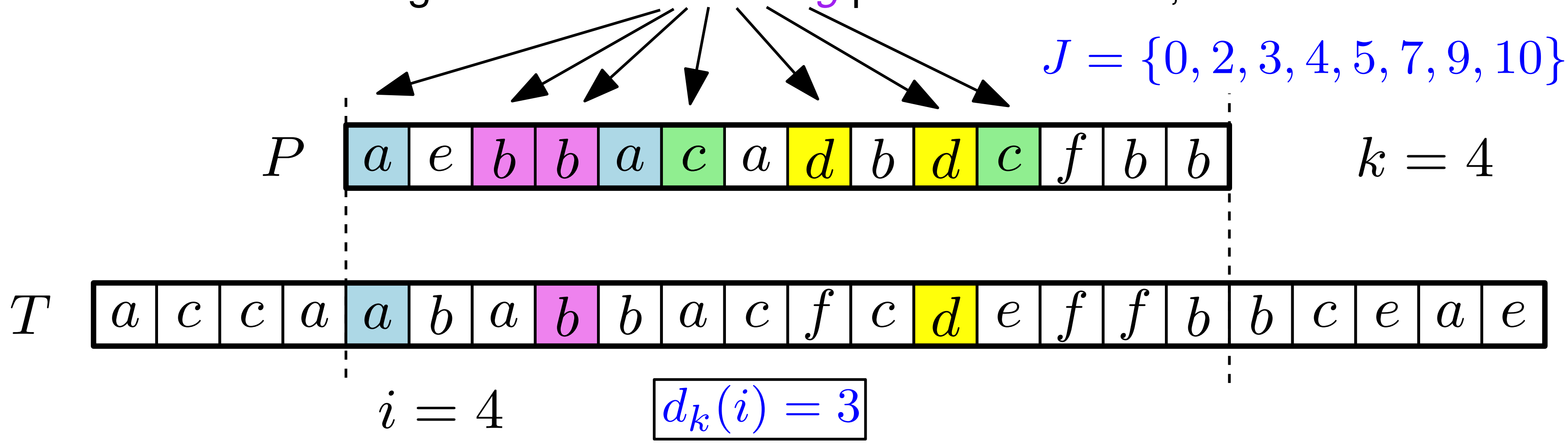
So $\sum_i d_k(i) \geqslant \frac{n}{\sqrt{k}} \cdot k > n\sqrt{k}$

Contradiction!

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



Let $d_k(i)$ be the number of $j \in J$ such that $P[j] = T[i + j]$
i.e. the number of (single character) matches involving interesting pattern locations

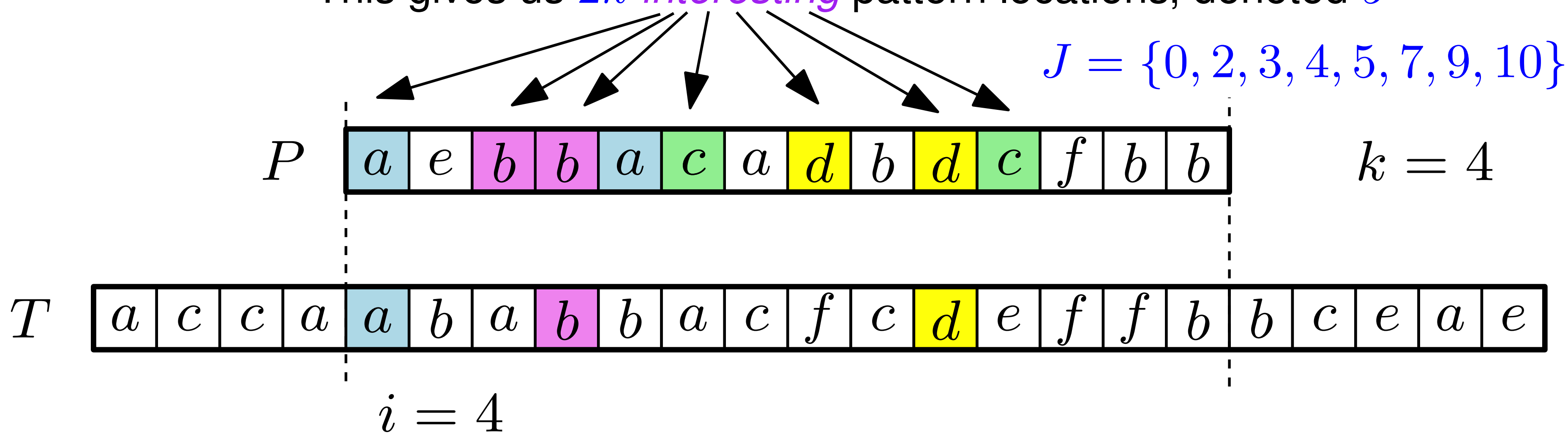
Fact if $d_k(i) < k$ then there are more than k mismatches (i.e. $\text{Ham}_k(i) = X$)
because there are $2k$ interesting positions... and fewer than k of them match

Fact There are at most $n/\sqrt{k}$ values of i with $d_k(i) \geqslant k$
this follows from a counting argument

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



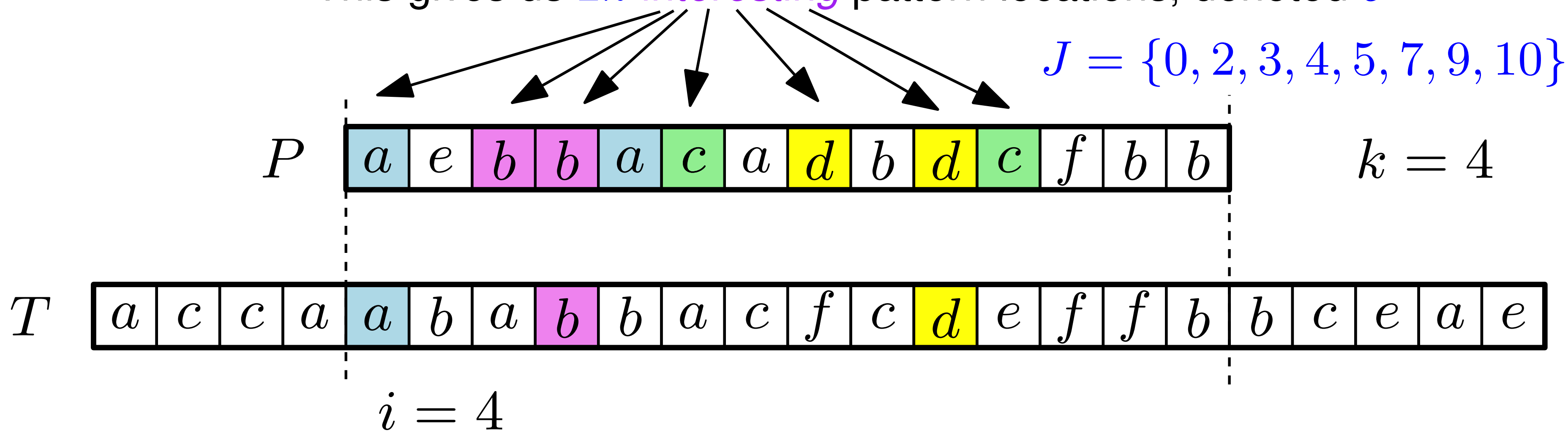
We can filter the text, leaving only $n/\sqrt{k}$ locations to check

every other location has more than k mismatches

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



We can filter the text, leaving only $n/\sqrt{k}$ locations to check

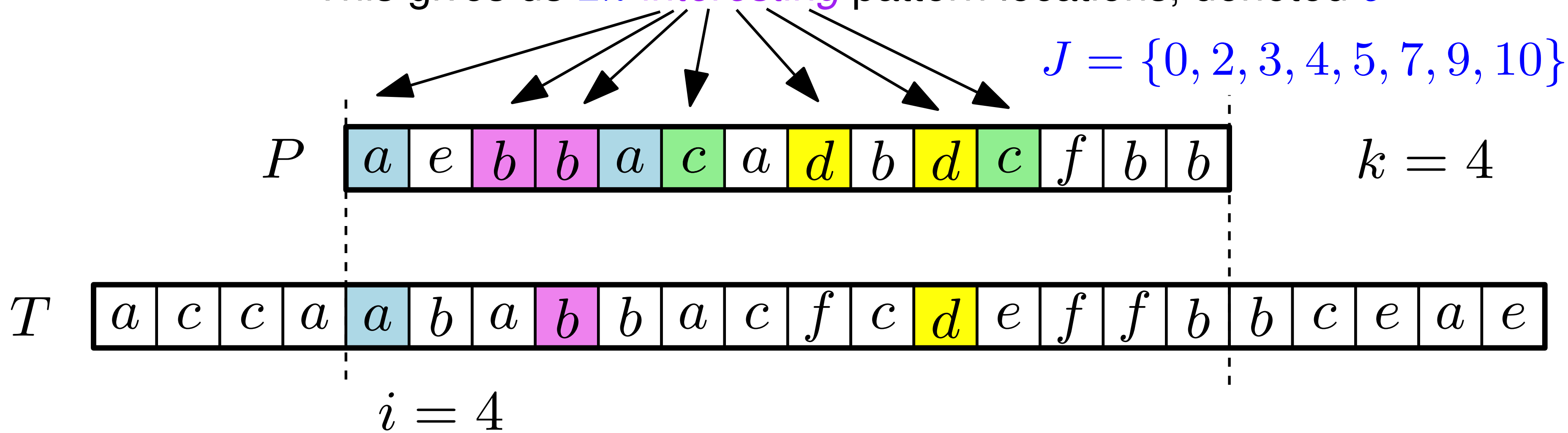
every other location has more than k mismatches

Check each of the remaining locations using LCP queries in $O(k)$ time

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



We can filter the text, leaving only $n/\sqrt{k}$ locations to check

every other location has more than k mismatches

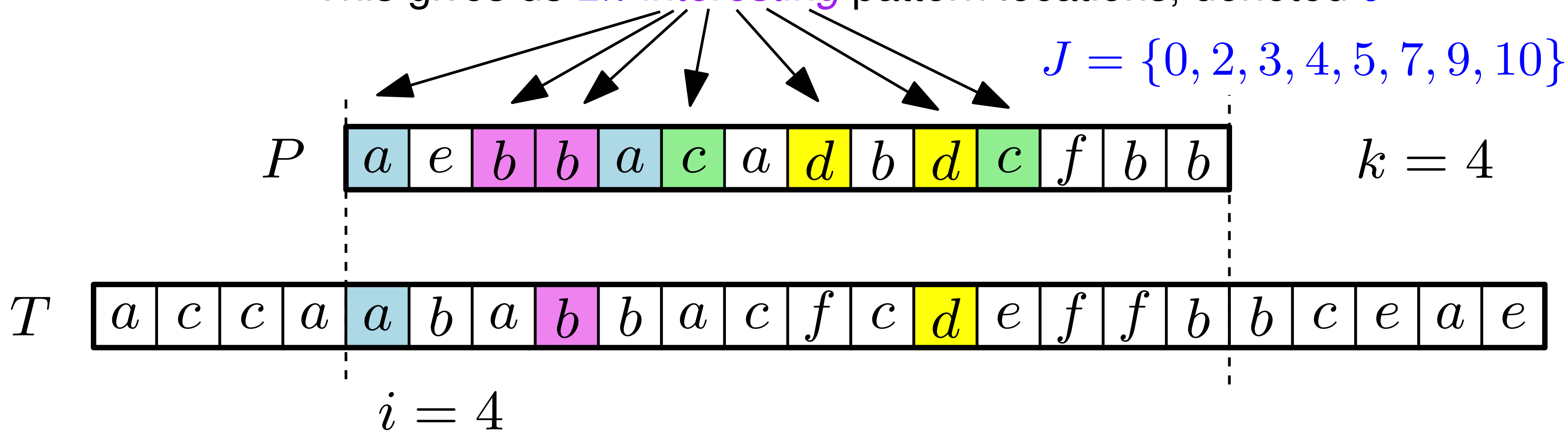
Check each of the remaining locations using LCP queries in $O(k)$ time

Determining which locations to check also takes $O(n\sqrt{k})$ total time

Case 2: There are at least $2\sqrt{k}$ frequent symbols

Pick any $2\sqrt{k}$ frequent symbols and for each symbol pick $\sqrt{k}$ occurences in P.

This gives us $2k$ *interesting* pattern locations, denoted J



We can filter the text, leaving only $n/\sqrt{k}$ locations to check

every other location has more than k mismatches

Check each of the remaining locations using LCP queries in $O(k)$ time

Determining which locations to check also takes $O(n\sqrt{k})$ *total time*

This gives $O(n\sqrt{k})$ total time

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P **has at most** $2\sqrt{k}$ **frequent symbols**

Case 2: P **has more than** $2\sqrt{k}$ **frequent symbols**

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P has at most $2\sqrt{k}$ frequent symbols

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Case 2: P has more than $2\sqrt{k}$ frequent symbols

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P has at most $2\sqrt{k}$ frequent symbols

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Count matches with infrequent symbols directly - $O(n\sqrt{k})$ time

Case 2: P has more than $2\sqrt{k}$ frequent symbols

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P has at most $2\sqrt{k}$ frequent symbols

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Count matches with infrequent symbols directly - $O(n\sqrt{k})$ time

Case 2: P has more than $2\sqrt{k}$ frequent symbols

Filter the text, leaving $n/\sqrt{k}$ alignments - $O(n\sqrt{k})$ time

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P has at most $2\sqrt{k}$ frequent symbols

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Count matches with infrequent symbols directly - $O(n\sqrt{k})$ time

Case 2: P has more than $2\sqrt{k}$ frequent symbols

Filter the text, leaving $n/\sqrt{k}$ alignments - $O(n\sqrt{k})$ time

Count mismatches at these alignments using LCP queries - $O(n\sqrt{k})$ time

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P **has at most** $2\sqrt{k}$ **frequent symbols**

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Count matches with infrequent symbols directly - $O(n\sqrt{k})$ time

Case 2: P **has more than** $2\sqrt{k}$ **frequent symbols**

Filter the text, leaving $n/\sqrt{k}$ alignments - $O(n\sqrt{k})$ time

Count mismatches at these alignments using LCP queries - $O(n\sqrt{k})$ time

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

Pattern matching with k-mismatches: putting it all together

Algorithm summary

Preprocess P, T for LCP queries - $O(n)$ time

Count the number of *frequent* symbols in P - $O(m \log m)$ time

Case 1: P has at most $2\sqrt{k}$ frequent symbols

Count matches with frequent symbols using convolution - $O(n\sqrt{k} \log m)$ time

Count matches with infrequent symbols directly - $O(n\sqrt{k})$ time

Case 2: P has more than $2\sqrt{k}$ frequent symbols

Filter the text, leaving $n/\sqrt{k}$ alignments - $O(n\sqrt{k})$ time

Count mismatches at these alignments using LCP queries - $O(n\sqrt{k})$ time

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

- this can be improved to $O(n\sqrt{k \log k})$