A_zval reflection

Time

Learning Objectives met

Lecture Helpfulness

What was bad about the assignment?
A_zalg due today!

Remember you can re-use code from a_zval

You should not use code from the internet!
Exact Pattern Matching w/ Z-algorithm

Pattern, $P$  \hspace{1cm} Text, $T$

Naive $\approx \theta(|P| \cdot |T|)$  \hspace{1cm} Z-Algorithm $\approx \theta(|P| + |T|)$

Find instances of $P$ in $T$

‘instances’: An exact, full length copy
Why continue?

The Z-algorithm is:

The Z-algorithm is: \( O(|P| + |T|) \) time

An alphabet-independent solution

The Z-algorithm is less good at:

Searching for a \textbf{set} of patterns (Aho-Corasick)

Running in \textit{sub-linear}* time (Boyer-Moore)

* — in practice, not theory
Exact pattern matching \(\textit{w/} \ \text{Boyer-Moore}\)

Boyer Moore \textbf{preprocesses} the pattern

\[ P \rightarrow \text{Preprocess} \rightarrow T \rightarrow \text{Boyer-Moore} \approx O(|P| + |T|) \]

Find instances of \(P\) in \(T\)

‘instances’: An exact, full length copy

\[ \approx O(|P|) \]
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{cat} \]

\[ T: \text{carl carried the cat} \]

\[ \text{cat} \]

\[ \text{carl carried the cat} \]

\[ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ldots \]

What does this alignment tell us?
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{cat} \]

\[ T: \text{carl carried the cat} \]

\[ \text{car} \ldots \text{cat} \]

\[ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ \ldots \]

What does this alignment tell us?

1) Our pattern doesn’t match at this alignment

There is no ‘r’ in ‘cat’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{ cat} \]
\[ T: \text{ car l c a r r i e d t h e c a t} \]
\[ \text{cat} \]
\[ \theta 1 2 3 4 5 6 7 8 9 \ldots \]

What does this alignment tell us?

2) Our pattern doesn’t match at *later* alignments

\[ \text{cat} \quad \text{car} \]

There is no ‘r’ in ‘cat’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{cat} \]

\[ T: \text{carl carried the cat} \]

\[ \text{cat} \]

\[ \emptyset 1 2 3 4 5 6 7 8 9 \ldots \]

What does this alignment tell us?

2) Our pattern doesn’t match at *later* alignments

- There is no ‘r’ in ‘cat’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{cat} \]
\[ T: \text{carl carried the cat} \]
\[ \text{cat} \]
\[ \text{cat} \quad \text{skip!} \]
\[ \text{cat} \quad \text{skip!} \]

What does this alignment tell us?

2) Our pattern doesn’t match at *later* alignments

\[ \text{car} \quad \text{There is no ‘r’ in ‘cat’!} \]
\[ \text{cat} \]
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[P: \text{word} \]

\[T: \text{There would have been a ...} \]

\[\ldots \text{word} \ldots\]

\[0123456789\ldots\]
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{word} \]

\[ T: \text{There would have been a \ldots} \]

\[ \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \ldots \\
\hline
\text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} & \text{word} \\
\end{array} \]

1) Our pattern doesn’t match at this alignment

\[ T: \text{woul} \]

\[ P: \text{word} \]
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{word} \]

\[ T: \text{There would have been a ...} \]

\[ \text{word} \]

\[ \text{0123456789...} \]

How many alignments can we skip?

2) Our pattern doesn’t match at *later* alignments

\[ T: \text{woul} \]

\[ P: \text{word} \]

There is no ‘u’ in ‘word’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{word} \]
\[ T: \text{There would have been a ...} \]

---

How many alignments can we skip? 2

2) Our pattern doesn’t match at *later* alignments

\[ T: \text{woul} \]
\[ P: \text{word} \]

There is no ‘u’ in ‘word’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{word} \]

\[ T: \text{There would have been a ... word word word word word word word word skip! word word word skip! word word} \]

How many alignments can we skip? \[2\]

2) Our pattern doesn’t match at *later* alignments

\[ T: \text{woul} \]

\[ P: \text{word} \]

There is no ‘u’ in ‘word’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{T A G A C} \]

\[ T: \text{G T A G A T G G C T G A T C G A G T A G C G G C G} \]

- \text{T A G A C}

How many alignments can we skip? 3

There IS a T in ‘TAGAC’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: T A G A C \]
\[ T: G T A G A T G G C T G A T C G A G T A G C G G C G \]

---

How many alignments can we skip? 3

There IS a T in ‘TAGAC’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \ A \ A \ B \ B \ B \]
\[ T: \ A \ A \ A \ B \ A \ B \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \ A \]
\[ \quad \vdash \ A \ A \ B \ B \ B \]

How many alignments can we skip? 1

AABAB

There IS an A in ‘AAABB’!
Boyer-Moore

**Intuition:** Learn from alignments to avoid others

\[ P: \text{A A B B B} \]

\[ T: \text{A A A B A B A A A A A A A A A A A A A A A A} \]

\[ \text{A A B B B} \quad \text{skip!} \]

\[ \text{A A B B B} \quad \text{the first match we encounter!} \]

How many alignments can we skip? \textbf{1}

AAB\textbf{AB}  
AAB\textbf{BB}  

There IS an A in ‘AAABB’!
Boyer-Moore: Bad Character rule

Upon mismatch, skip alignments until (a) mismatch becomes a match, or (b) $P$ moves past mismatched character. (c) If there was no mismatch, don't skip.
Boyer-Moore: Bad Character rule

Step 1:

- **T:** CCTTCTGCTACCTTTTGCGCGCGCGC
- **P:** CCTTTTGC
- Skip!

Step 2:

- **T:** CCTTCCTGCTACCTTTTGCGCGCGCGC
- **P:** CCTTTTGCG

Step 3:

- **T:** CCTTCCTGCTACCTTTTGCGCGCGCGC
- **P:** CCTTTTGCG
- Skip!

We skipped three alignments

Can we do anything to make this better?
Boyer-Moore: Bad Character rule

Which of the following alignments skips the most?

A)
\[ T: \text{TATAT...} \]
\[ P: \text{TAGAC} \]

B)
\[ T: \text{TTGAT...} \]
\[ P: \text{TAGAC} \]

C)
\[ T: \text{TAGAT...} \]
\[ P: \text{TAGAC} \]

D)
\[ T: \text{TAGTT...} \]
\[ P: \text{TAGAC} \]
Boyer-Moore: Bad Character rule improvement

Continue to test alignment from left-to-right

\[ P: \text{T A G A C} \]

\[ T: \text{G T A G A T G G C T G A T C G A G T A G C G G C G} \]

\[ \text{T A G A C} \]

... but compare \textit{characters} from right to left.
Right-to-left-scanning w/ BC Rule

\[ P: \text{word} \]

\[ T: \text{There would have been a ... word} \]

How many alignments do we skip?

There is no ‘l’ in ‘word’!
Right-to-left-scanning w/ BC Rule

**P:** word

**T:** There would have been a ...

How many alignments do we skip? 3
Right-to-left-scanning w/ BC Rule

Upon mismatch, skip alignments until (a) mismatch becomes a match, or (b) $P$ moves past mismatched character. (c) If there was no mismatch, don't skip.
Right-to-left-scanning w/ BC Rule

Step 1:

T: CCTTCTGCTACCTTTTGGCGCGCGCGGGAA
P: CCTTTTGGC

Step 2:

T: CCTTCTGCTACCTTTTGGCGCGCGCGGGAA
P: CCTTTTGGC

Step 3:

T: CCTTCTGCTACCTTTTGCGCGCGCGGGAA
P: CCTTTTGGC

Up to step 3, we skipped 8 alignments

5 characters in $T$ were never looked at
Right-to-left-scanning w/ BC Rule

Learn from character comparisons to skip pointless alignments

1. When we hit a mismatch $c$, move $P$ along until $c$ becomes a match (or $P$ moves past $c$) "Bad character rule"

2. Try alignments in one direction, but do character comparisons in *opposite* direction "Right-to-left scanning"

How do we put the first two rules in practice?

Exact pattern matching \textbf{w/ Boyer-Moore}

Boyer Moore \textbf{preprocesses} the pattern

\begin{align*}
\text{Preprocess} & \quad \approx O(\mid P \mid) \\
\text{Boyer-Moore} & \quad \approx O(\mid P \mid + \mid T \mid)
\end{align*}

Find instances of $P$ in $T$

‘instances’: An exact, full length copy
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: \ T\ C\ G\ C \) \( \Sigma: \ A\ C\ G\ T \)

The goal is to produce a table which tracks *skips*
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: \text{T C G C} \quad \Sigma: \text{A C G T} \)

The goal is to produce a table which tracks **skips**

<table>
<thead>
<tr>
<th>( \Sigma )</th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: \) T C G C \( \Sigma: \) A C G T

The goal is to produce a table which tracks *skips*

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

\( P: \) T C G C

Boyer-Moore: BC rule preprocessing
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: T C G C \quad \Sigma: A C G T \)

The goal is to produce a table which tracks \textit{skips}

\[
\begin{array}{|c|c|c|c|}
\hline
\Sigma & T & C & G & C \\
\hline
A & & & & \\
C & & & & \\
G & & & & \\
T & & & 2 & \\
\hline
\end{array}
\]

\( T: ??? A ??? ??? \quad P: T C G C \)
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: $P: T C G C$, $\Sigma: A C G T$

The goal is to produce a table which tracks skips

\[
\begin{array}{cccc}
\Sigma & T & C & G & C \\
A &   &   &   & 3 \\
C &   &   &   & \\
G &   &   &   & \\
T &   &   &   & 2 \\
\end{array}
\]

$P: T C G C$
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: TCGC \) \( \Sigma: ACGT \)

The goal is to produce a table which tracks *skips*

\[
\begin{array}{cccc}
P & T & C & G & C \\
\hline
\Sigma & \Sigma \\
A & 0 & 1 & 2 & 3 \\
C & 0 & - & 0 & - \\
G & 0 & 1 & - & 0 \\
T & - & 0 & 1 & 2 \\
\end{array}
\]
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: $P$: B A B A A A B $\Sigma$: A B
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args:  

\[ P: B \ A \ B \ A \ A \ A \ B \]  

\[ \Sigma: A \ B \]

For each character \( p \) in pattern \( P \)

For each character \( c \) in alphabet \( \Sigma \)

Find the closest previous instance of \( p \) (to the left of \( c \)).
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: B A B A A A B \) \( \Sigma: A B \)

For each character \( p \) in pattern \( P \)

For each character \( c \) in alphabet \( \Sigma \)

Find the closest previous instance of \( p \) (to the left of \( c \)).

<table>
<thead>
<tr>
<th>( \Sigma )</th>
<th>B</th>
<th>A</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Boyer-Moore: BC rule preprocessing

Preprocessing requires two args: \( P: B \ A \ B \ A \ A \ A \ A \ B \) \hspace{1cm} \( \Sigma: A \ B \)

For each character \( p \) in pattern \( P \)

For each character \( c \) in alphabet \( \Sigma \)

Find the closest previous instance of \( p \) (to the left of \( c \)).

<table>
<thead>
<tr>
<th>Pattern</th>
<th>B</th>
<th>A</th>
<th>B</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Assignment 4: a_bmoore

Learning Objective:

Implement preprocessing of patterns with Boyer-Moore*

Observe Boyer-Moore* efficiency as a heuristic

Due: February 21th 11:59 PM

Consider: Optimal preprocessing is $\theta(|P| |\Sigma|)$. Can you code it?
Boyer-Moore: Using the BC Table

Try alignments from left-to-right and match characters from right-to-left

When we encounter a mismatch, skip the calculated number of alignments
Boyer-Moore: Using the BC Table

Try alignments from left-to-right and match characters from right-to-left

When we encounter a mismatch, skip the calculated number of alignments

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Σ

$T$: G G G G G G G G G G
$P$: T C G C
Boyer-Moore: Using the BC Table

Try alignments from left-to-right and match characters from right-to-left

When we encounter a mismatch, skip the calculated number of alignments

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Σ

$T$: A A T C A A T A G C
$P$: T C G C
Boyer-Moore: Tracking total skips

<table>
<thead>
<tr>
<th>Σ</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$T$: B B B B

$T$: B B B B B

$T$: B B B B B
Boyer-Moore: Tracking total skips

\[ \Sigma \]

\[ \begin{array}{ccc}
A & A & A \\
A & 0 & 0 & 0 \\
B & 0 & 1 & 2 \\
\end{array} \]

\[ P \]

\[ T: B \ B \ B \ B \]
Assignment 4: a_bmoore

Learning Objective:

Implement preprocessing of patterns with Boyer-Moore*

Observe Boyer-Moore* efficiency as a heuristic

Due: February 21th 11:59 PM

Consider: Our Boyer-Moore is theoretically slower than Z-algorithm. But is it slower in practice? What is our total character comparisons?