122COM: Boyer-Moore Searching

David Croft

Coventry University david.croft@coventry.ac.uk

2018



String searching better



- Naive search works but is inefficient.
- Obvious solution is not always the best one.
- Think about the problem and what is being searched.
 - Can you be smarter?



Boyer-Moore string searching algorithm.

- **1**977.
- Not going to talk about the whole algorithm here.
 - Gets complex.
- Right to left comparison.
- Can skip sections of the text.
 - Don't need to test every position.
 - How?



Boyer-Moore string searching algorithm.

- **1977.**
- Not going to talk about the whole algorithm here.
 - Gets complex.
- Right to left comparison.
- Can skip sections of the text.
 - Don't need to test every position.
 - How?
- Pre-processes the search string.
 - Produce bad character rule table.
 - Will explain how in a minute.



Boyer-Moore algorithm



- 1 Preprocess the search string to create the "bad character table".
 - Will explain how in a minute.

- Same at the naive search, position the search string at the start of the main text.
- Compare the strings.
- If strings don't match then in the bad character table lookup the character positioned at the end of the search string.
- Move the search string by the number of positions specified in the table.
- 6 Repeat from step 3.





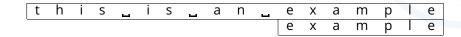
















- For each character.
 - Except the last.
- Just count number of places between it and end of search string.

example
$$\Rightarrow$$
 a e l m p x *



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.

example
$$\Rightarrow \frac{a \ e \ l \ m \ p \ x \ *}{4}$$



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.

example
$$\Rightarrow \frac{a \quad e \quad l \quad m \quad p \quad x \quad *}{4 \quad 6}$$



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.

example
$$\Rightarrow \frac{a \quad e \quad l \quad m \quad p \quad x \quad *}{4 \quad 6 \quad 1}$$



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.

example
$$\Rightarrow \frac{a \quad e \quad l \quad m \quad p \quad x \quad *}{4 \quad 6 \quad 1 \quad 3}$$



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.



- For each character.
 - Except the last.
- Just count number of places between it and end of search string.





Doesn't need to sort or modify the sequence being searched.

■ Small amount of pre-processing on the search value.

Worst case.

Linear time.

Average case

Sub-linear.

Not the only improved string searching algorithm.

- Knuth-Morris-Pratt.
- Finite State Machine (FSM).
- Rabin-Karp.





The End

