Some Hints for Lab1

Dave Ogle

Question

- If you are going to build a house, which of the following is the better first step?
    1. Go to Lowes to buy lumber
    2. Work with an architect to plan out the house, understand how it fits together

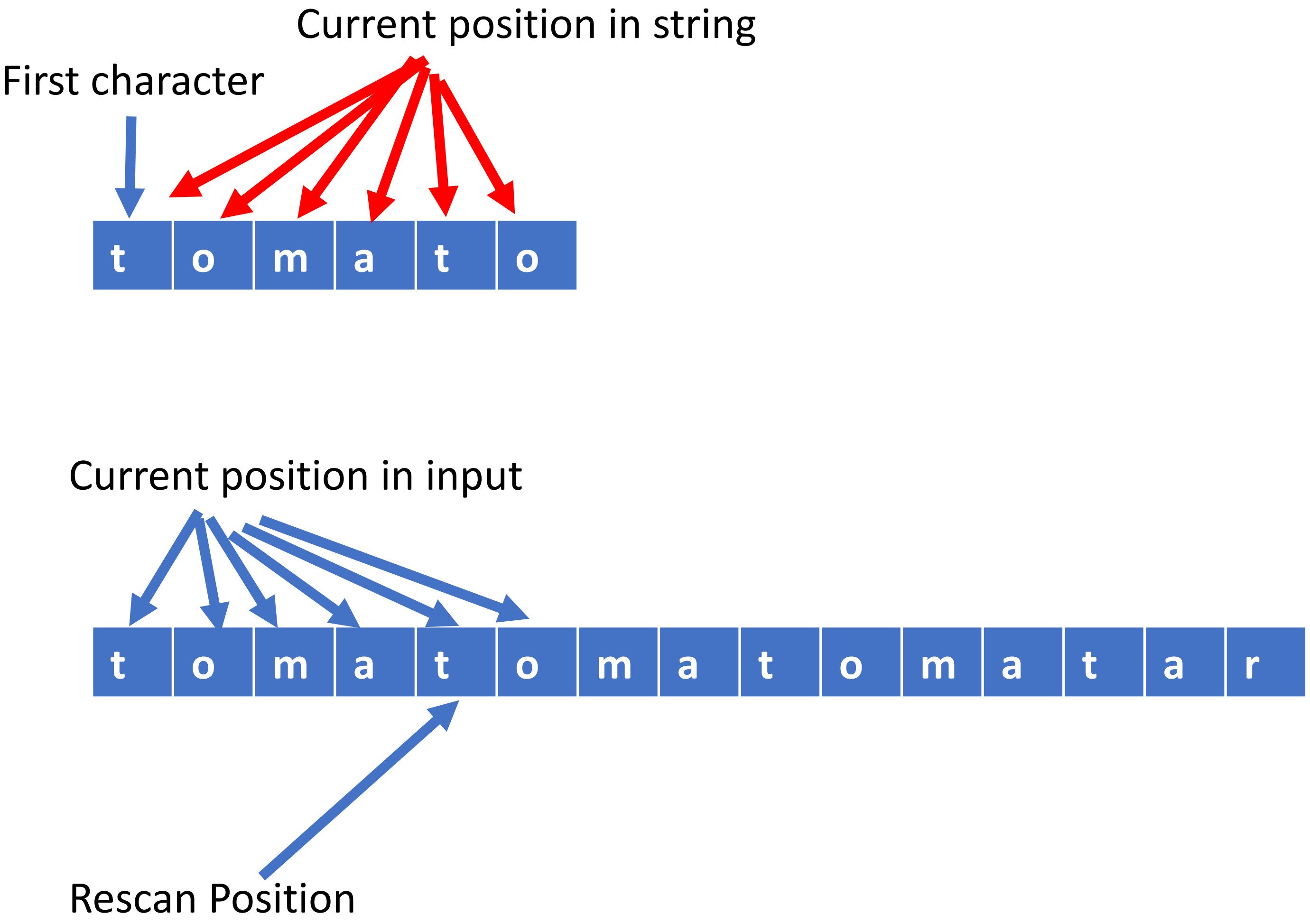
- If you answered #1, you’re in trouble, but…..
    - Most SW engineers immediately want to start coding

- Play around with binary or hex input
    - Figure out how to compare two characters/strings that are binary/hex

String Matching

- Consider the following example:
    - You are given the search string “tomato”
    - You are given the input of “tomatomatomator”
    - How many matches do you see?
    - The answer should be 3. Think about how you figured that out, what process did you use in your head.
    - Write those steps down
- Do not read the entire input file into memory
    - Input files might contain terabytes or petabytes of data.
    - Think about how many characters you need to have in memory at any one time, compared to the size of the search string
- Efficiency is not an issue for this lab
    - Think about the tradeoffs between memory usage, and file IO

One VERY inefficient way to solve the matching problem



Current position in string

First character

t	o	m	a	t	o

Current position in input

t	o	m	a	t	o	m	a	t	o	m	a	t	a	r

Rescan Position

Possible Match = YES!

Found Match = YES!

Comparing Characters

- You may find if you are not careful, that your application works with ‘text’ files but fails when trying to use binary files (like the jpg files)

- One trick I find works well – to only look at a ‘byte’ of something try to ‘and’ the char with hex ‘ff’
- Example
  - if (buffer[0] == (searchChar & 0xff))