* **What is String Matching**
* String Matching Problem is to find the match word or character provided by the user
* We can say that string matching we can find particular pattern in text.
* Usually, text editing program use string matching.
* It can be also use in Database schema, Network systems.
* **How to approach the problem**
* Let first think that string is a pattern- an array of character
* Text is an array T[1,2…n] where n is length
* Also we an array of pattern P[1….m] where m <= n.
* There is a counter (s) that is shift through the array of Text and find valid occurrence of P.
* Goal is to find all the valid shift (s) in the text T where pattern is P occurs.
* **Common Term to get familiar with**
* ⅀-\* = the set of all finite-length strings formed using characters from the alphabet Sigma.
* The zero-length ***empty string***, denoted €, also belongs to ⅀\*.
* So, all the P and T elements of ⅀\*
* When you combine two strings is denoted to xy.
* X=wy is the symbol for concatenating w and y
* The length of x is equal to the length of w+ the length of y
* **Naive- brute-force algorithm**

**Diagram

Description automatically generated**

* Text of length n is T[1…n]
* Pattern(P) of length m is P[1..m]
* Where m <= n.
* Where we say that elements of P and T are from Finite alphabet ⅀.
* Our purpose is to find all the occurrence of P in text T that is called valid shift(s). It could be more than one shifts.
* In the example above P occurs with shift s in text T or (or, equivalently, that pattern P *occurs beginning at position* s + 1 in text T ) if

0<= s <= n-m and

T [s+1..s+m] = P [1..m]

* In this example P occurs in T only once. And it calls valid shift(s). that is at shift s=3.
* **There are four algorithms for string matching**

**Table

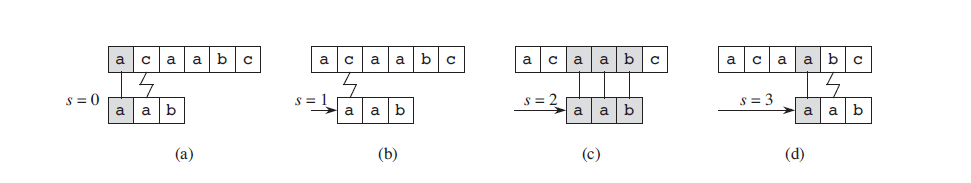
Description automatically generated**

* **Naive string-matching algorithm**
* Here is pseudo code for Naive string algorithm

Text

Description automatically generated

* Running time for this algorithm is 0(n-m+1)m).
* So let me explain more about this algorithm. Line 3 will check the condition of s if that is true than it goes in to for loop and do the if statement. That will check if P==T if that is true than it will print out where shift has occurred.
* Here is sliding a “template” Example
* In this algorithm slides the pattern one by one and check at each shift if character match or not if it match than it will print the match shift.



* So as you can see in first slide it compare a to a it matches the character so it will match second character it doesn’t match. Now we start from 2 shift and see if it match when it match it prints out and see if we have any other occurrence.
* **The Rabin-Karp Algorithm**
* Like the Naive Algorithm, Rabin-Karp algorithm also slides the pattern one by one. But unlike the Naive algorithm, Rabin Karp algorithm matches the hash value of the pattern with the hash value of current substring of text, and if the hash values match then only it starts matching individual characters. So Rabin Karp algorithm needs to calculate hash values for following strings.  
  1) Pattern itself.   
  2) All the substrings of the text of length m.
* Here is example

Chart, scatter chart

Description automatically generated

* In the example above we convert character to decimal and we let just think that ⅀ ={1,2…9} in decimal
* Each character is a digit in radix-d notation, where d=|⅀|
* The Rabin-Karp algorithm calculate a hash value for the pattern and for each M-character subsequence of text to be compared.
* It will check if the hash value is same or not. If hash value is different than it will calculate the value for next M-character.

Table

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

* As you can see in this example it keeps testing to insure check all the possibilities.
* **Rabin-Karp example (Alphabets)**
* Lets say for this example we have alphabet with lets say 8 letters.
* Alphabet = q,b,c,d,…,h
* And just think that for this example we say that alphabet “a” value is 1 and “b” value is 2 so on and so forth.
* So as an example what would be hash value for “aed”

Diagram

Description automatically generated

* **Running time**

Text, letter

Description automatically generated

* Worst case

Theta(m) + Theta (n-mm+1)\*m) = Theta ((n-m+1)\*m)

* It check all the possibilities
* Best Case Theta(m) +theta (n-m+1) = Theta(n-m+1)

This happened when you don’t have any match

* **String Matching with finite automata**
* What is Finite automata
* The string-matching automaton is a very useful tool which is used in string matching algorithm. It examines every character in the text exactly once and reports all the valid shifts in O (n) time. The goal of string matching is to find the location of specific text pattern within the larger body of text

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

Here Q is = {0,1}

qo = 0

accepting state A Q is = 0,1

⅀ = a,b

Diagram

Description automatically generated

* Now lets see another example

Graphical user interface, text, application

Description automatically generated

A picture containing text, device, meter

Description automatically generated

A picture containing text, meter, device, close

Description automatically generatedA picture containing text, device, meter, control panel

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

* Where T is Input text
* is = Transition function
* Accepting state
* Processing time for this is
* **The Knuth Morris-Pratt algorithm**
* The **Knuth-Morris-Pratt (KMP) algorithm** is an algorithm that is used to search for a substring (W), in a given string (S), in O(m+n)*O*(*m*+*n*) time (where m*m* and n*n* are the lengths of W and S).

A picture containing diagram

Description automatically generated

A picture containing text

Description automatically generated

Text, letter

Description automatically generated

Text, letter

Description automatically generated

* Question 1: what is matching time for naive algorithm?
* Answer; O((n – m = 1)m)
* Question 2: at what shift you will find pattern P=001 in text T= 0101001000
* Answer: Shift = 4 we will find the matching pattern.
* In robin -Karp algorithm after it find the match algorithm stops and doesn’t need to check if we have any other possibilities.
* False
* Running time of Robin-Karp algorithm if we don’t have any match
* Theta(n-m+1)
* Which algorithm is mostly use in hardware design
* Answer: Finite automata