Naive string matching

~~we are~~ given a string of characters and ~~a pattern~~

Pattern matching is ~~the~~ Comparing two or more text or array of Characters. It is also called string matching

Naive string matching

* No preprocessing is done
* Comparison is done in left to right order on the Characters in two arrays.
* When mismatch occurs pattern is moved one position to the right w.r.t ~~the~~ text.
* Again Comparison is done from left to right

Inputs are text T and pattern to be matched P

Algo

```
n ← length [T]
m ← length [P]                          ← (n-m+1) times
for s=0 to s=n-m
{
    Counter = 0
    for i=0 to m-1                      ← m times
    { if (P[i] == T[s+i])
          Counter ← Counter+1;
    }
    if (counter = m)  // Pattern matched
    Print (s)
}
```

Complexity = $(n-m+1) * m$

$= O(m.(n-m+1))$

ex: T = 0000 1000 101001

P = 0001

s=0 no match

s=1 T = 0000 1000 101001

P = 0001

matched at s=1

s=2 T = 0000 1000 101001

P = 0001

No match

s=3 T = 0000 1000 101001

P = 0001

No match

s=4 T = 0000 1000 101001

P = 0001

No match

s=5 T = 0000 1000 101001

P = 0001

s=5 matched

s=6 T = 0000 1000 101001

P = 0001

No match

s=7 T = 0000 1000 101001

P = 0001

No match

s=8 T = 0000 1000 101001
P = 0001
No match

s=9 T = 0000 1000 101001
P = 0001
no match

s=10 T = 0000 1000 101001
P = 0001
no match

s=11 T = 0000 1000 101001
P = 0001
Pattern matched at s=11

So, Pattern matched at pos 1, 5, 11