Alex Josué Flórez Farfán
Magister en Ciencias de la Computación

PROGRAMACIÓN COMPETITIVA

Ciencia de la Computación – UNSA
Segundo semestre 2021

TERCERA UNIDAD

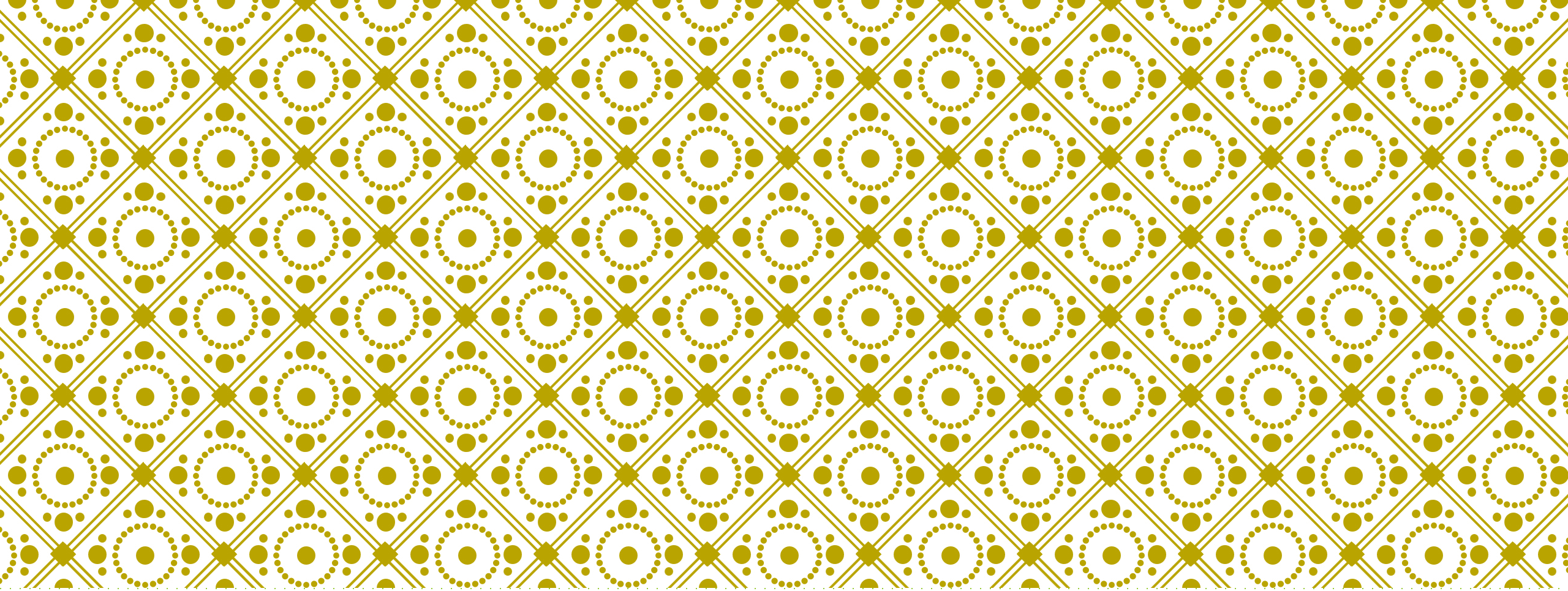
Nota Evaluación Continua

Repositorio

- Directorios para cada aula
- Nombres de archivos

Participaciones en clase

Examen



STRINGS

1. SUBSTRING ORDER I

https://cses.fi/problemset/task/2108

Given a string of length n.

If all of its distinct substrings are ordered lexicographically, what is the kth smallest of them?

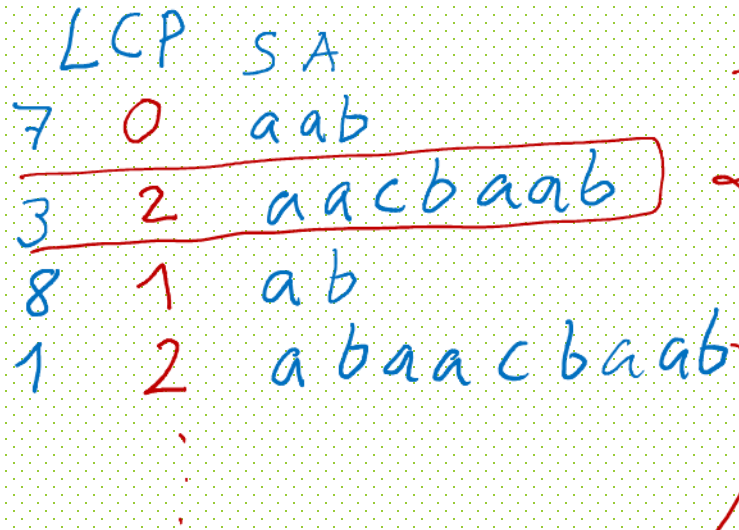
Input:	Output:
babaacbaab	aba
10	

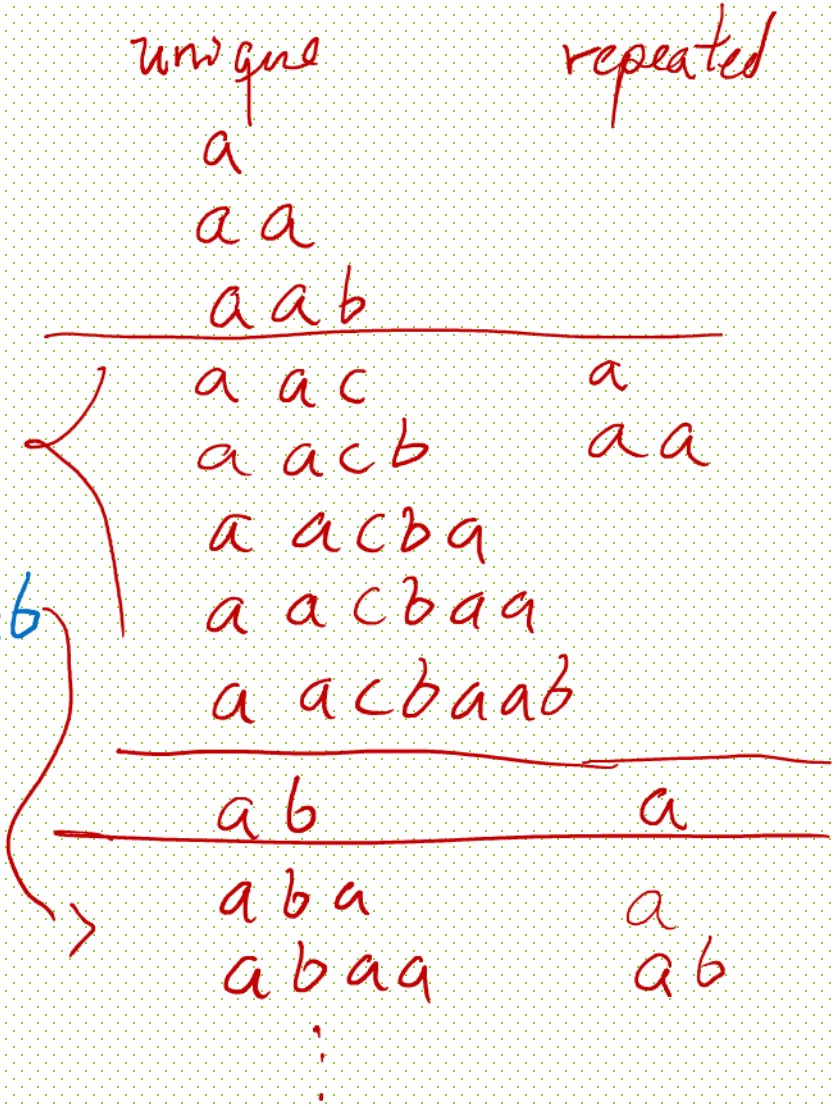
The 10 smallest distinct substrings in order are

a, aa, aab, aac, aacb, aacba, aacbaa, aacbaab, ab, and aba.

1. SUBSTRING ORDER I

0 1 2 3 4 5 6 7 8 9
babaacbaab
8 3 7 1 4 9 6 0 2 5





2. SUBSTRING ORDER II

https://cses.fi/problemset/task/2109

Given a string of length n.

If all of its substrings (not necessarily distinct) are ordered lexicographically, what is the kth smallest of them?

Input:

```
baabaa
10
```

Output:

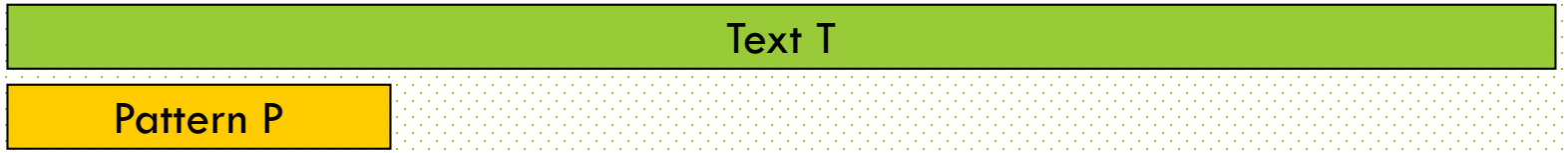
```
ab
```

The 10 smallest substrings in order are

`a, a, a, a, aa, aa, aab, aaba, aabaa, and ab`

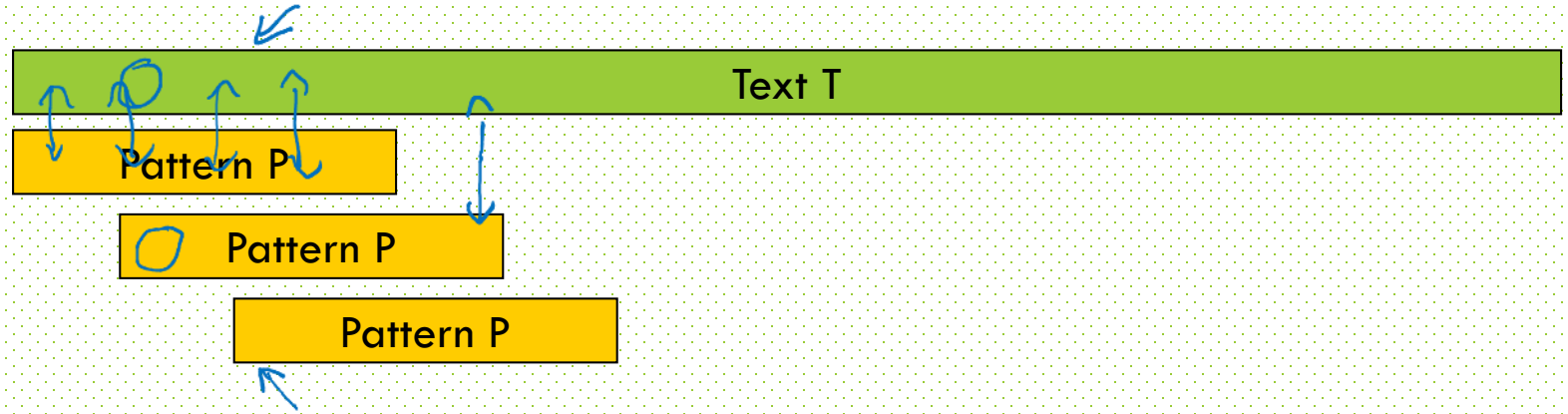
3. STRING MATCHING

Given a string T, the problem of string matching deals with finding whether a pattern P occurs in T and if P does occur then return the position in T where P occurs.



NAÏVE ALGORITHM (OR BRUTE FORCE)

Assume $|T| = n$ and $|P| = m$



Compare until a match is found.
If so return the index where match occurs else return -1

O(m*n)

KNUTH MORRIS PRATT

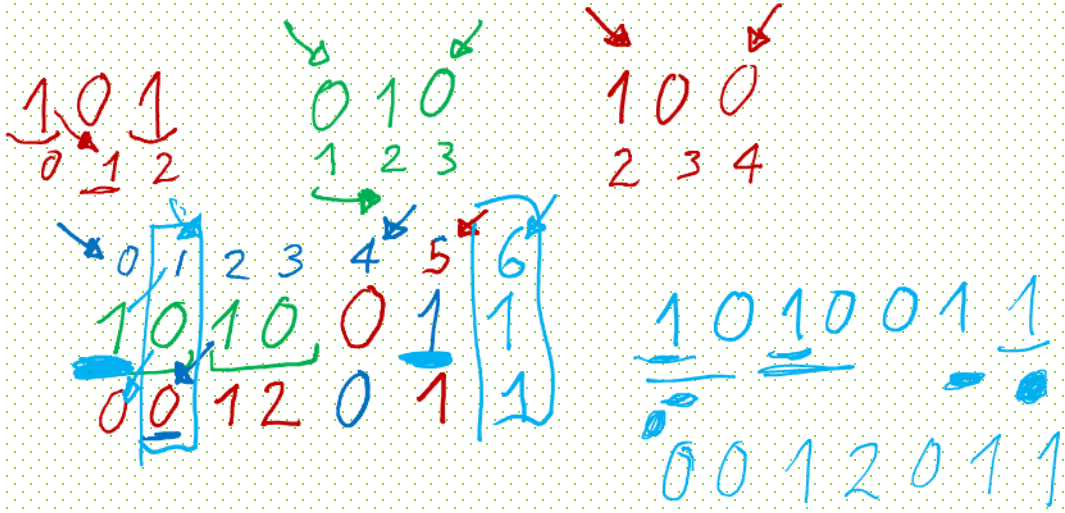
KMP

Compute in advance how far to jump in P when a match fails.

P = 1010011

0 1 2 3 4 5 6

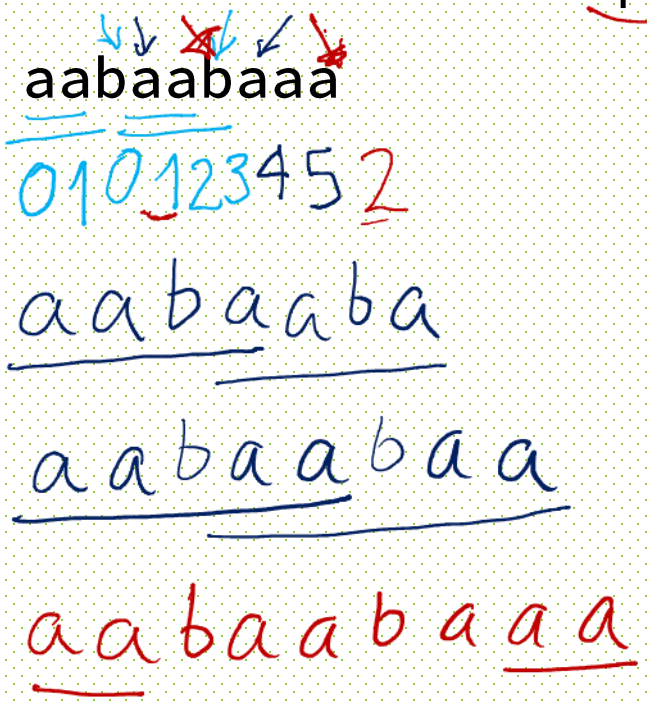
ab x ab



KNUTH MORRIS PRATT

Compute in advance how far to jump in P when a match fails.

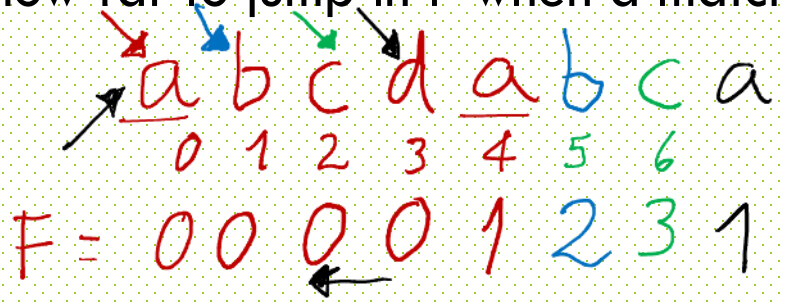
P = aabaabaaa



KNUTH MORRIS PRATT

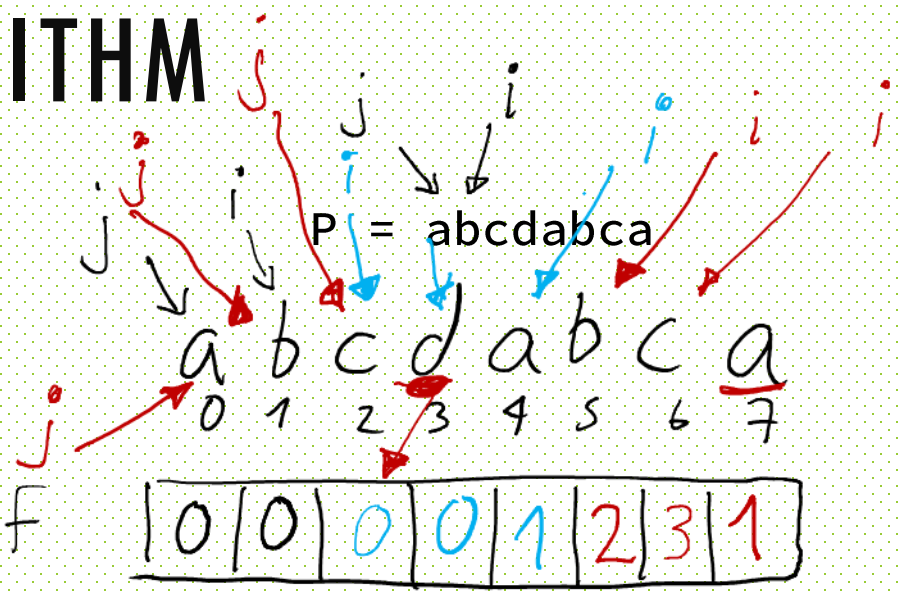
Compute in advance how far to jump in P when a match fails.

P = abcdabca



KMP PRE-PROCESS ALGORITHM

```
 m = |P|;
Define a table F of size m
F[0] = 0;
i = 1; j = 0;
while(i<m) {
   compare P[i] and P[j];
   if(P[j]==P[i])
     { F[i] = j+1;
       i++; j++; }
   else if (j>0) j=F[j-1];
   else {F[i] = 0; i++;}
}
```



KNUTH MORRIS PRATT

T = abxabcabcaby

P = abcdabca

F = 00001231

KNUTH MORRIS PRATT

T = abxabcabcaby

P = abcaby

F = 000120

KMP ALGORITHM

```
 input: Text T and Pattern P
|T| = n
|P| = m
Compute Table F for Pattern P
i=j=0

while(i<n) {
  if(P[j]==T[i])
    { if (j==m-1) return i-m+1;
      i++; j++; }
  else if (j>0) j=F[j-1];
  else i++;
}
output: first occurrence of P in T
```

$O(m)$

$O(n)$

$O(m+n)$

3. STRING MATCHING

https://open.kattis.com/problems/stringmatching

For each test case, output one line containing the positions of all the occurences of pattern in text, from first to last, separated by a single space.

```
p
Popup
helo
Hello there!
peek a boo
you speek a bootiful language
anas
bananananaspaj
```

```
2 4

5
7
```