Course Instructor: Prof. Amihood Amir **Scribe: Poorva Malviya**

Deterministic Sampling

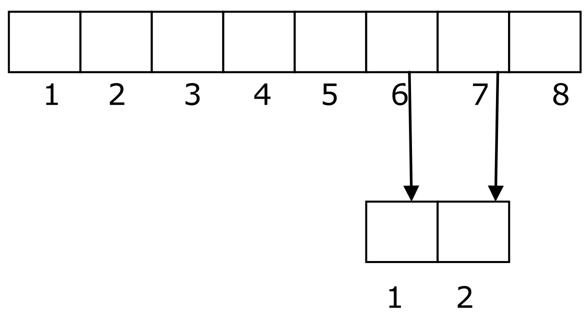
The deterministic sampling Idea

If a non-periodic pattern is given and a small deterministic sample (DS) is generated such that if sample positions do not match with Text positions then there is no occurrence of Pattern in Text.

The sample DS with size *s* is an ordered set where l<=(logm -1). And Deterministic sample is DS =[ds(1), ds(2), d(3),…d(j),…, ds(l)].

Pattern

DS



In Example, the pattern has length 8, hence size of DS<= 2 i.e. (log8-1)

In Text matching, for every location 1<=i<=n-m+1, occurrence of DS is checked.

<u>Case 1:</u> No occurrence of DS found

Result: No occurrence of Pattern in Text

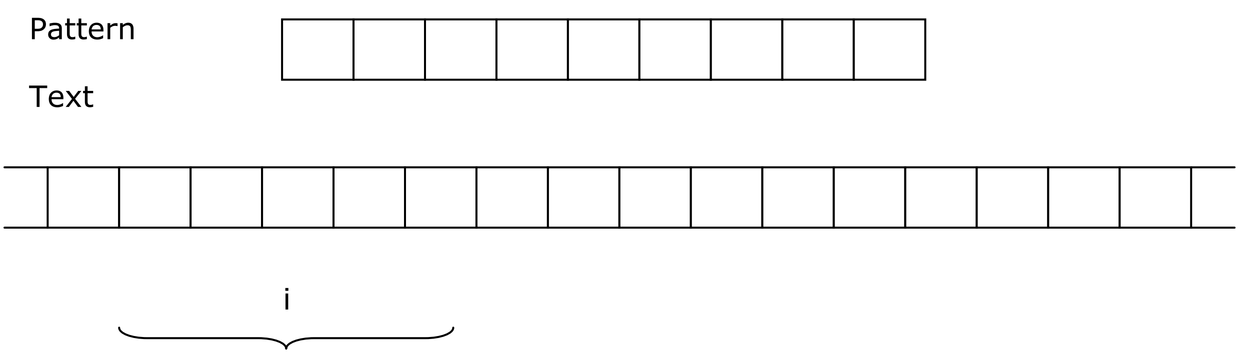
<u>Case 2:</u> Occurrence of DS found for candidate location i

Result: Let x be the index of sample's start position in Pattern. Then based on the candidacy of location i, candidates in the locations x-1 preceding i and m/2 – x locations succeeding i can be eliminated. This property is known as Ricochet property of Deterministic samples.

The location i-x+1 through i-1 and i+1 through i +m/2 –x constitutes dead zone.

Pattern

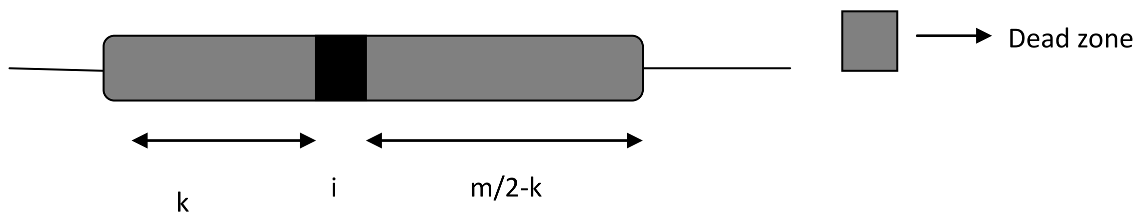
Text



Every occurrence of DS guarantees a dead zone of length = m/c where c depends on x.

Hence characteristics of good DS:

1. Length of DS is small. |DS| = O(logm)
2. There exists an integer k such that if DS occurs at position i in the text then no occurrence of pattern starts in section [i-k…… i+m/2 -k] except i. This section [i-k ……. i+m/2 -k] except i is dead zone



Pattern Matching Algorithm

<u>STEP 1</u>: Get candidate positions using DS

```
count ← 0                                    / /maintains the count of candidates
for  i← 1 to n-m+1
        match ← true
        for  j← 1 to s
                if T[i+j+x] ≠ DS[j]
                        match ← false
                end
        end
        if match = true
                count =count+1
                Candidate[count] = i
        end
end
```

STEP 2: Remove Candidates from the list which lie in dead zone

```
 n ⟵ 1
RemainingCandidates[1] = Candidates[1]
for m⟵  1 to Length[Candidates] -1
       if Candidates[m+1] – Candidates[m] >x
              n =n+1
              RemainingCandidates[n] = Candidates[m+1]
       end
end
```

STEP 3: Use naïve based pattern matching approach for RemainingCandidates.

Time complexity

Step 1 takes O(n*s) or O(nlogm)

Step 3 verification of remaining candidates (n/(m/c)) takes O(m* (nc/m)) or O(nc)

Hence the total Time complexity is O(nlogm).

VISHKIN (1990)

For non-periodic pattern s a DS of size logm can be constructed in linear time.

Periodicity

A String P= U^kU' is said to be periodic if k>1 and U' is a prefix of U. So U is the period of P.

Example:

ABABABA is periodic because it can be represented as (AB)^3A

ABA is not periodic as in representation $(AB)^1A$ K=1

Let us take an example of a periodic String

P= ABCDABCDABCDABCDABCDABCDABCDABCDAB

P can be represented as :

$P = (ABCDABCDABCDABCD)^2AB$

Period is (ABCDABCDABCDABCD)

Or

$P = (ABCDABCD)^4AB$

Period is (ABCDABCD)

Or

$P = (ABCD)^8AB$

Period is (ABCD)

All the above representations are valid periodic representations.

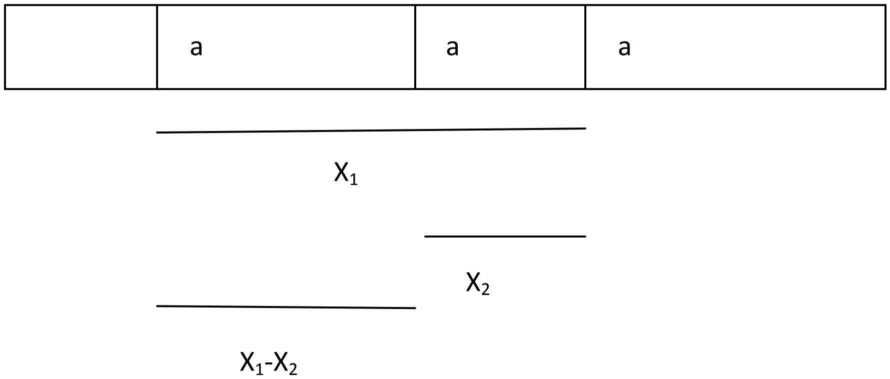
Alternate view of Periodicity

Let length of P is m and length of U is x then for $P = U^kU'$

$P[i] = P[i+x]$

Where $1 <= i <= m-x$

Periodicity Lemma



Let U_1, U_2 be periods of P and let $|U_1|$ = x_1 and $|U_2|$= x_2

Then, P has a period U_3 where $|U_3|$= gcd($x_{1,}$ x_2)

Claim : If two numbers b and c are co-prime and b>c then b-c and c are co-prime

Proof : Let us assume that b-c and c are not co-prime, hence they have a common factor x besides 1

Therefore, b-c = x .w

and c = x.z

On adding b-c and c, we get

b = x.w +x.z =x.(w+z)

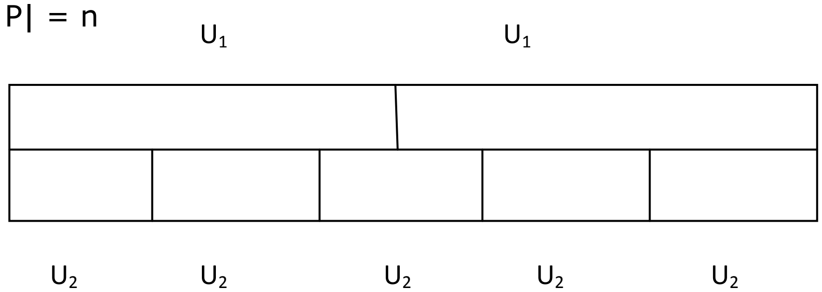
This implies that b and c share a common factor x, which contradicts our assumption that b and c are co-prime.

Proof of Periodicity lemma by Induction

For |P| = 1 is obvious

Let us assume that periodicity lemma hold for all the strings of length < n

For |P| = n



Consider U_1, $x_1 > x_2$, hence $U_1[i] = U_1[i+x_2]$ where $1<=i <= x_1-x_2$

As U_2 is also a period of P

Hence, $P[i] = P[i+x_1] = P[i+x_1 - x_2]$ where $1<=i<=m-x_1$

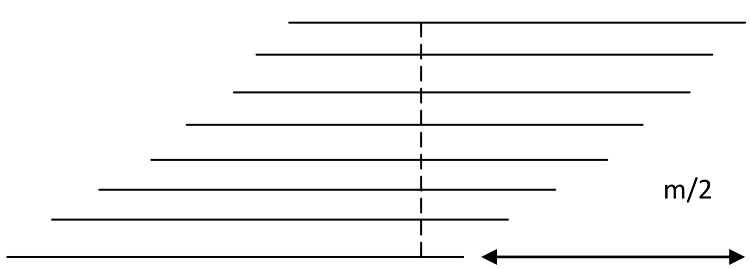
This concludes that U_1 has a period of length $(x_1 - x_2)$.

From Induction hypothesis we can say U_1 has a period of length

$\gcd(x_1 - x_2, x_2)$. But as per the claim, this is equal to $\gcd(x_1, x_2)$, hence P also has a peiod of length $\gcd(x_1, x_2)$

Sample construction

Assume that we have the witness table WIT for the pattern. Let us consider P shifted and stacked m/2 times. If a line is drawn at a position j then it can intersect i-th row or not. If the line intersects i-th row then symbol (i,j) is present at the intersection.



Claim 1: If i1 and i2 be two different elements of P[1….m/2] then there exists an integer j such that j-th column intersects both i1 and i2 with symbol(i1, j) ≠ symbol(i2, j). j can be obtained from WIT in constant time.

Due to non-periodicity, for occurrences of pattern placed at i1 and i2, there is a mismatch at position j given by j = i2 + WIT[i2-i1]

Claim2: If J is a set of rows and if a vertical colum intersects the first and the last row of J then it intersects all the rows of J.

Procedure:

1. Choose column j where symbol a occurs less than half times
2. Discard all the rows in which symbol a isn't present at column j
3. Repeat until only one row is left

Example-

```
        A B A B B A A B A B
      A B A B B A A B A B
    A B A B B A A B A B
  A B A B B A A B A B
A B A B B A A B A B
```

```
      A B A B B A A B A B
A B A B B A A B A B
1 2 3 4 5 6 7 8 9 10
```

Sample : 7, A

8, B

Pattern matching in periodic patterns

Step 1: Find the smallest period of the pattern

Step2: Partition the text into windows of m/2 . For each window, if there are more than 2 occurrences of sample then consider just first and last occurrences as possible candidates for occurrence of pattern.

Step 3: Check all the possible candidates in naïve way.