KMP Algo

T abcdefg

P def

def

3 shift def

def

Pattern abcdabc

prefix a, ab, abc, abcd, abcda, abcdab

suffix c, bc, abc, dabc, cdabc, bcdabc

is there any prefix which is suffix

we have to find longest prefix which is suffix also.

Π table / LPS

a	b	c	d	a	b	e	a	b	f	
0	0	0	0	1	2	0	1	2	0	

a	b	c	d	e	a	b	f	a	b	c
0	0	0	0	0	1	2	0	1	2	3

a	a	a	a	b	a	a	c	d	
0	1	2	3	0	1	2	0	0	

Girish Vishwanath Ishaq Snehal

table

a	a	a	a	b
0	1	2	3	0

a	b	a	b	a	c	a
0	0	1	2	3	0	1

m = len(Pattern)

m = 4

j = 0 i = 1.

(i < m)

if Pat(0) = Pat(1)

a = a yes

j ++

Pi[1] = j

Pi[1] = 1.

i++ i = 2

Pat(1) = Pat(2)

a = a yes

j++ j = 2

Pi[i] = j

Pi[2] = 2.

i++ = 3

Pat[3] = Pat[j]

a = a yes

j++ = 3

Pi[i] = j

3

i++ ⇒ 4

Pat[i] = Pat[j]

a ≠ b

if (j ≠ 0)

j = 0

j = Pi[j-1]

j = Pi[2]

j = 2.

if Pat[3] = Pat[j]

Pat[4] = Pat[2]

b ≠ a

if (j ≠ 0)

j = Pi[j-1]

j = Pi[1]

j = 1

Pat[i] = Pat[j]

b ≠ a No

j ≠ 0

j = Pi[j-1]

= Pi[0]

j = 0

Pat[i] = Pat[j]

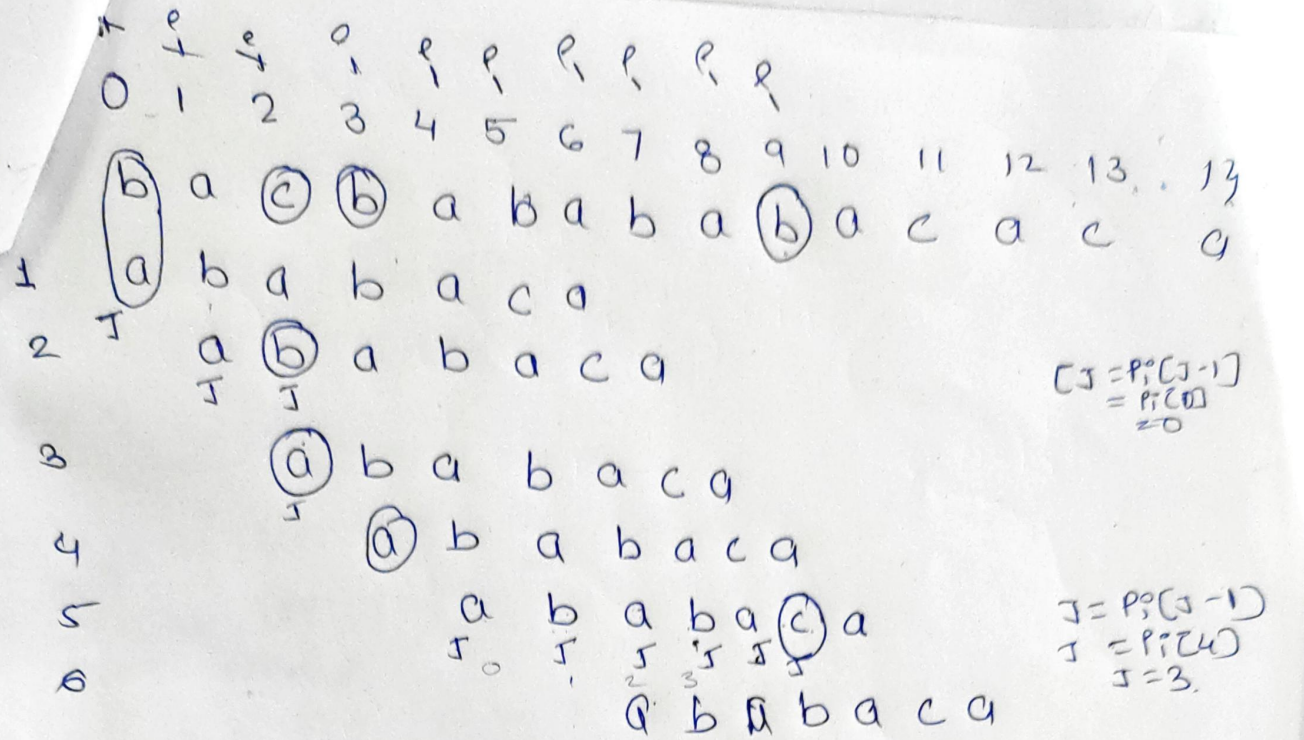
b ≠ a No

j = 0 ∴

Pi[5] = 0

Pi[4] = 0

i++



1. Pat(i) ≠ Pat(J)
   So i++ (Now i = 1, J = 0)

2. Mismatch at
   i = 2, J = 1
   Pat[i] ≠ Pat[J]
   & J ≠ 0
   So J = Pi[J-1] = Pi[0]
   J = 0
   so Now i = 2, J = 0

3. Mismatch at
   i = 2 & J = 0
   i++ i = 3, J = 0

4. mismatch
   i = 3 & J = 0
   i++ Now i = 4, J = 0

5.) Mismatch at i = 9 & J = 5

So update J
J = Pi[J-1]
= Pi[4]
J = 3
Now i = 9, J = 3

6. i = 9, J = 3
   Matched

Pattern :-

a	b	a	b	d
0	0	1	2	0

n = len txt = 14 Pattern
m = len Pat. 6

J	0	1	2	3	4	5	6
	a	b	a	b	a	c	a
	0	0	1	2	3	0	1

eg

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	b	a	c	b	a	b	a	b	a	b	a	c	a	c	a
	a	b	a	b	a	c	a		b	a	b	a	c	a	
J	0	1	-	-	-	-	-								

if Pat[J] = txt[i]
Pat[0] = txt[0]
Not equal
J != 0
J = 0
i++.

J ≠ 0
J = Pi[J-1]
= Pi[0]
J = 0

How = i is at c
J is at a

at J = 5 mismatch

Pat[J] != txt[i]
c ≠ b.

J ≠ 0
J = Pi[J-1]
= Pi[4]
J = 3.

CLASS TEST / MID - SEMESTER / PRELIMINARY EXAMINATION

TERM I / II - 20 - 20

Supervisor's Signature & Date

Name :	Roll No.:
Year :	Branch :
Subject :	Date :
Section :	Marks :

Concept of Randomized & Approximation Algo

- NP Complete prob. can not be solved in Polynomial time

Such prob. can be solved using either Randomized algo or Approximation Algo within Polynomial time complexity and Polynomial space complexity.

Randomized Algorithms :-

Consider an application. It may be possible to solve this application in various way out of which only one way is the best.

To find the best Method, we may have to follow all the various approaches and then select the best. But in some situation It may be desired that- Instead of wasting more time to find the best approach, it is better to select- another approach randomly which is near to best-. This saves the time and gives result- earlier.

Q.No.					

Randomized Algorithm is an algorithm which makes use of randomizer.

Characteristics of Randomized Algorithm:-

When a Randomized algo A run using an Input I more than once, then it may generate different o/p each time. This characteristics is benifical when there are more than one correct ans.

2 types of Randomized Algo :-

(1) Monte Carlo Algo

(2) Las Vegas Algo

(1) Monte Carlo :- it finds correct ans. most of the time with a high probability whatever the instance is.

The probability of generating wrong ans. is very less

But when the algo generates wrong ans. no warning is given usually

Monte carlo algo is said P-correct if it to finds a correct solut^n with at least P proba. for any instance.

Q.No.					

(2) Las Vegas Algo :- it always finds a correct ans. it may take more time to find correct solve for good i/p but may take average time to solve for the bad i/p

Adv. of Randomized Algo :-

(i) Simplicity :- Randomized algo are generally found to be simpler as compared to the deterministic algo to solve the same prob

(ii) Efficiency or speed :- Randomized algo are much faster as they have been shown to provide better complexity bounds.

(3) ~~Paradigms~~ :-

Disadvantage of Randomized Algo :-

Performance :- it is possible that the randomized algo may or may not improve the performance of an algo.

Probability of Error - for critical appli" small prob. of error or small prob. of requiring time to run algo is not acceptable.

Ex :- Nuclear reactor.

Q.No.

Randomized Sorting Algo :-

① Consider quick sort algo :

when the quick sort algo is solved using deterministic algo then Divide and Conquer strategy is used. it require $O(n \log n)$ time in best and avg case. but it takes $O(n^2)$ time in worst case. even if all the ele. in the list are already sorted.

Randomized Quick sort algo can be design using Las Vegas approach -

Algo takes list of n ele. to be sorted and o/p is sorted elements

Algorithm Random Quick Sort (L)

i/p - List L of n ele. to be sorted
o/p - Sorted List

```
begin
    if small (L)
      print sorted list (L)
    else
        x = select Uniform Pivot (List)
        L1 = { y | y∈L and y<x }
        L2 = { y | y∈L and y≥x }
        Random Quick Sort (L1)
        Print x
        Random Quick Sort (L2)
    end if
end.
```