Danyl Fernandes 2020012004 (72) 26-04-2021

AOA Experiment 10

Aim:

To implement & analyze Rabin-Karp Algorithm:

Implementation:

```
public class RabinKarp {
     public final static int d = 256;
     static void search(String pat, String txt, int q) {
     int M = pat.length();
     int N = txt.length();
     int i, j;
     int p = 0;
     int t = 0;
     int h = 1;
     for (i = 0; i < M - 1; i++)
           h = (h * d) % q;
     for (i = 0; i < M; i++) {
           p = (d * p + pat.charAt(i)) % q;
           t = (d * t + txt.charAt(i)) % q;
     }
     for (i = 0; i \le N - M; i++) {
           if (p == t) {
                for (j = 0; j < M; j++) {
```

```
if (txt.charAt(i + j) != pat.charAt(j))
                      break;
                }
                if (j == M)
                System.out.println("Pattern found at index " + i);
           }
           if (i < N - M) {
                t = (d * (t - txt.charAt(i) * h) + txt.charAt(i + M)) % q;
                if (t < 0)
                t = (t + q);
           }
     }
     }
     public static void main(String[] args) {
     String txt = "DANYL LOVES LINUX";
     String pat = "LINUX";
     int q = 13;
     search(pat, txt, q);
     }
}
```

Output:

```
C:\Users\thearchhero\.jdks\openjdk-15.0.2\bin\java.
Pattern found at index 12

Process finished with exit code 0
```

	Daniel Faranches
	Danyl Fernandes 2020012004 (72)
	2-2-00 (2-01 (+2)
	Exp 10
	Theory:
-	Rabin I kaup proposed a string - matching
	algorithm that performs well in practice & that also generalizes to other
-	& that also generalizes to other
	algorithms for related problems such as two-dimensional pattern natching
	as two-dimensional pattern natching
	CT: 11 - marker was all alexanders
	number - the ontic notions want to refer.
	to the control was the transmission
	to the equivalence of two numbers modulo a third number.
	module a proper
M 1	Like the Naive Alsonthm, Rabin-Kalp
	Like the Naive Algorithm, Rabin-Kalp Algorithm also slides the pattern one by one But unlike the naive approach, Rabin-Kalp Algo. matches the hash
	one But unlike the name approach
	Rabin - Kaip Algo. matches the hash
	values of the pattures with the hash
	values of the pattures with the hash values of unevent substrays of the
	text .
	If there is a match found, only then
	it begiss matching individual character
	As, such, Rabin - troup algo needs to
	Calculate hash values for following
	i) Patten itself
	2) All the substrings of the text
~~	al lands

Danyl Femandes 2020012004 (72) Algorthm: Rabin_ Koup_Matcher (T, P, d, q) n = T. length m = P. lengthh = dm-1 mod q for i = 1 to m p = (dp + P[i]) modg to = (dto + T[i]) modg for s= 0 to n-m if p == ts if P[1...m] == T[s+1...s+m] print "Pattern occurs with if scn-m ts+1=(d(ts-T[s+1]h)+ T[s+m+1]) mody Analysis: The average & best cose sunning time of the Rabin-Kaep algorithm is O(n+m) but its worst-time is O(nm). Worst Cose of Rabin - kap algorithm occurs when all characters of pattern & text are same as the host values of all the substrings of txt[] match with hash value of patt[]. For example patt[] = "AAAAAAA".

	Danyl Fernandes 2020012004 (72)	3
	Example:	ssume
	Prampic.	a = 0
(m=3)	Pattern: a a c	b=1
	Mash: $1+1+2=4$	C= 2
- •	Mar Pelandella and house	d=3
	a born Fills = di	
(n=5)	TEXT: a a a a c	
	I1 1+1+1=3	
	I2 1+1+1=3	
	I3 1+1+2=4)	
	MAN THE PARTY OF T	
	Pallen	
	Pattern found	
14.	Tound Tound	
	C 1 Manager All All All All All All All All All Al	
	We successfully implemented & and	lyred
	Rabin-halp Algenthm.	J
1	rabin - reap my	
	Statistics of the second secon	
	e away & bat core suaponi tim	MTP
1 Crist	1 8. I well and a death midely with	<u> </u>
	its want time is a company with	
	and millions to that a probable to as	MAGE JA
++	in all the diamating at a politicing & it	
1100	same as the hash values of	9 100
	I'm Appea [] fat to shugams	
	specificated forter or the company	(a i)