Name : G. Nithish

Regno : 19BCS0012

Course : Design of algorithms

Course Code : CSC 3007

Date : 03/02/2021

① String matching Program to check whether the given pattern is matched in the Original string or not.

Source Code:

```
#include <iostream>
using namespace std;

int Stringmatch(string t, string P, int n, int m)
{
    int j, i;

for(i=0; i<n-m; i++)
 {
    j=0;
while(j<m && p[j]==t[i+j])
 { j++;
 }
```

```
    if(j==m)
      return i;
    }
      return -1;
}

    main()
{
    String text, Pattern;
    int i,j,n,m,found;
    Cout<<"Enter the text :";
    Cin>> text;
    Cout<<" Enter the Pattren :";
    Cin>> Pattren;

    n = text.length();
    m = Pattern.length();

found = stringmatch(text,Pattern,n,m);

if (found!=-1)
    cout<<"th string is found at
                the index :"<<found;
```

```
else
  cout << "the string is not faund in
                the given text";
}
```

Analysing the time Complexity of the program.

Lets take the Loop:-

```
for(i=0; i< n-m; i++)
  {
    j=0;
While (j<m && P[j] == t[i+j])
  {  j++;          ] ⟶ 'm' times
  }
if (j==m)
return i; }
```

(n-m+1) * m

→ Here the while condition execute 'm' time's the inner loop

→ With combination of both "for" and "while" loop it execute the inner statement by (n-m+1)*m is time Complexity of this Programe.

Given a pattern 'M' characters in length and a text 'N' characters in length

Worst Case:

Compare pattern to each substring of text of length M. for eg m = 3.

Eg.

① K K K K K K K K K K K K K K K K K K K K K K A → text

K K A

② K K K K K K K K K K K K K K K K K K K K K K A

K K A - → pattren

③ . . . . . .

. . . . . .

K K K K K K K K K K K K K K K K K K K K K K A

K K A

Total no. of comparisons : m (n-m+1)

Worst Case time Complexity: O(mn)

Best Case:

* Given a pattern 'm' characters in length and a text 'n' characters in length:

* Best Case if pattern found:

Finds pattern in first M positions of texts. for example "M=3"

① K K A K K K K K K K K → text

K K A → pattern

Only 3 comparisons made.

Total no. of comparisons : M

Best Case time complexity: O(M)

The Random or Average Case : $\Theta(n+m)$ time complexity

If the pattern not found or found at after many comparisons it comes under worst case:

② Code segment to depict O, Ω, Θ :-

Source code :-

```
#include<iostream.h>
int main()
{
   int i,j, n=5;          ---> 1 "for assignment statement".
   for(i=1; i<=n; i++)    ---> n+1 is time complexity
   {
      for(j=1; j<=n; j++)  ---> n times
      {
         cout<<j;          ---> n^2
      }
   }
   return 0;
}
```

time complexity

Total time complexity :- n^2+n+1

(O) upper bound :- $O(n^2+n+1) = O(n^2)$

(Ω) Lower bound :- $\Omega(n^2+n+1) = \Omega(1)$

(Θ) Average bound :- $\Theta(n^2+n+1) \Rightarrow \Theta(n)$

Asymptotic notation:-

Asymptotic notations are the expressions that are used to represent the complexity of an algorithm

Types:-

1.) Big-O notation(O)- Big O notation specifically describes worst case scenario.
(upper bound)

2.) Omeg Notation(Ω) - Omega (Ω) notation specifically describes best case scenario.
(lower bound)

3.) Theta Notation(Θ) - This notation represents the average complexity of an algorithm.
(Average bound)

3.) $15n^3 + 6n^2 + 5n + 3$ is $O(n^3)$

Soln

$$f(n) \in O(g(n))$$

if $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ where c and c are positive constants.

$f(n) = 15n^2 + 6n^2 + 5n + 3$ | $g(n) = n^3$

$f(n) \leq c \times g(n) \quad \forall\ n \geq n_0$

let us Assume $n_0 = 1$, $c = ?$

$\Rightarrow 15n^3 + 6n^2 + 5n + 3 \leq c \cdot n^3$ for all $n \geq 1$

$\Rightarrow 15 + 6/n + 5/n^2 + \frac{3}{n^3} \leq c$ for all $n \geq 1$

$15 + \frac{6}{1} + \frac{5}{1^2} + \frac{3}{1^3} \Rightarrow 15 + 6 + 5 + 3 \Rightarrow 29$

$\therefore 29 \leq c$ for all $n \geq 1$

So Let us assume $n_0 = 1$, $c \geq 29$

for eg. take $c = 30$

$\Rightarrow 15n^3 + 6n^2 + 5n + 3 \leq 30 \cdot n^3$ for all $n \geq 1$

$\div n^3$

$\Rightarrow 15 + 6/n + 5/n^2 + \frac{3}{n^3} \leq 30$ for all $n \geq 1$

$\Rightarrow \quad 29 \leq 30$ for all $n \geq 1$

$\therefore f(n) \in O(g(n)) \rightarrow 15n^3 + 6n^2 + 5n + 3 \in O(n^3)$