### Implement Horspool’s algorithm for String Matching. Time analysis to be done by varying the input size.

Code:

#include<iostream>

#include<string.h>

#include<math.h>

using namespace std;

int count = 0;

int horspool(string t,string p){

    int n =t.length();

    int m = p.length();

    int table[130];

    for (int i=0;i<126;i++){

        table[i] = m;

    }

    for(int j=0;j<=m-2;j++){

        table[p[j]] = m-1-j;

    }

    int i = m-1;

    while(i<=n-1){

    int k = 0;

    count++;

    while(k<=m-1 && p[m-1-k]==t[i-k])

    {

        k++;

        count++;

    }

    if(k==m)

     return i-m+1;

    else

     i = i+table[t[i]];

    }

    return -1;

}

int main(){

    // cout<<"Enter text : \n";

    // string t;

    // // cin>>t;

    // getline(cin,t);

    // cout<<"Enter pattern :\n";

    // string p;

    // // cin>>p;

    // getline(cin,p);

    char a[26] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g','h', 'i', 'j', 'k', 'l', 'm', 'n','o', 'p', 'q', 'r', 's', 't', 'u','v', 'w', 'x', 'y', 'z' };

    string t = "";

    cout<<"Enter length of text :";

    int n;

    cin>>n;

    for (int i = 0; i < n; i++)

        t = t + a[rand() % 26];

    cout<<"text :"<<t<<endl;

    cout<<"Enter pattern :\n";

    string p;

    cin>>p;

    int x = horspool(t,p);

// for worst case......

//     string t1,p1;

//     for(int i=0;i<n;i++)

//     {

//         t1[i]='0';

//     }

//

//     p1 = '1000';

//     int x = horspool(t1,p1);

    if(x>=0)

     cout<<"Pattern found at index "<<x<<endl;

    else

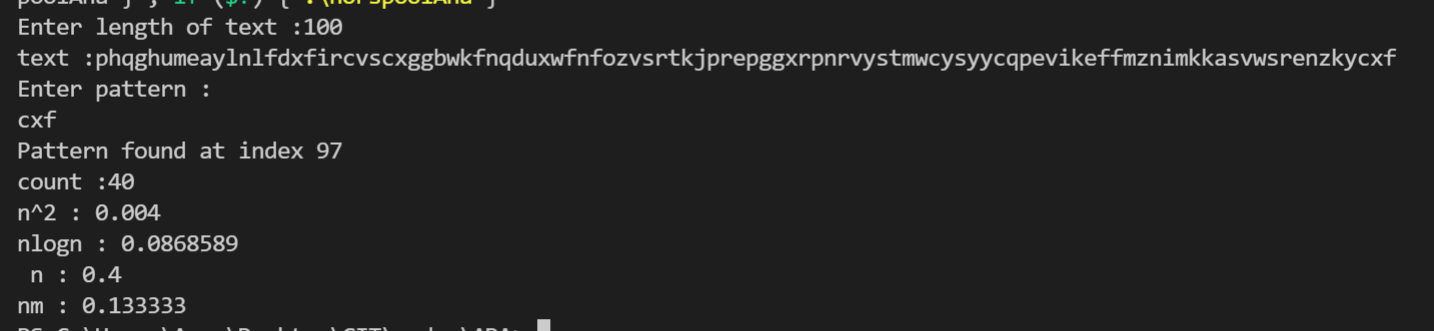
     cout<<"Pattern not found\n";

    cout<<"count :"<<count<<endl;

    cout<<"n^2 : "<<count/float(n\*n)<<"\nnlogn : "<<count/float(n\*log(n))<<"\n n : "<<count/float(n)<<"nm : "<<count/float(n\*3)<<endl;

}

Output:



### Analysis:

