Searching for Patterns | Set 1 (Naive Pattern Searching)

Given a text *txt[0..n-1]* and a pattern *pat[0..m-1]*, write a function *search(char pat[], char txt[])* that prints all occurrences of *pat[]* in *txt[]*. You may assume that $n > m$.

Examples:
1) Input:

```
txt[] =  "THIS IS A TEST TEXT"
pat[] = "TEST"
```

Output:

```
Pattern found at index 10
```

2) Input:

```
txt[] =  "AABAACAADAABAAABAA"
pat[] = "AABA"
```

Output:

```
Pattern found at index 0
Pattern found at index 9
Pattern found at index 13
```

Pattern searching is an important problem in computer science. When we do search for a string in notepad/word file or browser or database, pattern searching algorithms are used to show the search results.

Naive Pattern Searching:
Slide the pattern over text one by one and check for a match. If a match is found, then slides by 1 again to check for subsequent matches.

```
#include<stdio.h>
#include<string.h>
void search(char *pat, char *txt)
{
    int M = strlen(pat);
    int N = strlen(txt);

    /* A loop to slide pat[] one by one */
    for (int i = 0; i <= N - M; i++)
    {
        int j;

        /* For current index i, check for pattern match */
        for (j = 0; j < M; j++)
        {
            if (txt[i+j] != pat[j])
                break;
        }
        if (j == M)  // if pat[0...M-1] = txt[i, i+1, ...i+M-1]
        {
           printf("Pattern found at index %d \n", i);
        }
    }
}

/* Driver program to test above function */
int main()
{
   char *txt = "AABAACAADAABAAABAA";
   char *pat = "AABA";
   search(pat, txt);
   getchar();
   return 0;
}
```

What is the best case?
The best case occurs when the first character of the pattern is not present in text at all.

```
txt[]  = "AABCCAADDEE"
pat[] = "FAA"
```

The number of comparisons in best case is O(n).

What is the worst case ?
The worst case of Naive Pattern Searching occurs in following scenarios.
1) When all characters of the text and pattern are same.

```
txt[] = "AAAAAAAAAAAAAAAAAA"
pat[] = "AAAAA".
```

2) Worst case also occurs when only the last character is different.

```
txt[] = "AAAAAAAAAAAAAAAAAB"
pat[] = "AAAAB"
```

Number of comparisons in worst case is O(m*(n-m+1)). Although strings which have repeated characters are not likely to appear in English text, they may well occur in other applications (for example, in binary texts). The KMP matching algorithm improves the worst case to O(n). We will be covering KMP in the next post. Also, we will be writing more posts to cover all pattern searching algorithms and data structures.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Recursively print all sentences that can be formed from list of word lists
- Check if a given sequence of moves for a robot is circular or not
- Find the longest substring with k unique characters in a given string
- Function to find Number of customers who could not get a computer
- Find maximum depth of nested parenthesis in a string
- Find all distinct palindromic sub-strings of a given string
- Find if a given string can be represented from a substring by iterating the substring “n” times
- Suffix Tree Application 6 – Longest Palindromic Substring

Tags: Pattern Searching

Writing code in comment? Please use **ideone.com** and share the link here.

Join the discussion…

Madhup Pandey • 10 days ago

check this :
it is simple and takes less time and memory than any other pattern searching algo :

http://ideone.com/KfPQIL

• Reply • Share ›

Abhishek Jain • 18 days ago

Described here is an approach with the naive algorithm as well as the KMP (Knuth-Morris-Pratt) algorithm: http://www.devcodenote.com/201...

• Reply • Share ›

nemo • 2 months ago

this won't work if the pattern is of size = 1

• Reply • Share ›

Guest • 2 months ago

If some one is looking for java implementation :

public class NaivePatternMatcher {

/**

* This class helps you to find if a pattern exists in the given text.

*

* As an example lets consider the following.

*

* Given a text,

*

* T = "THIS IS A BIRD"

*

see more

• Reply • Share ›

Guest • 3 months ago

// Just checking

```
// just checking
int main()
{
return 0;
}
```

• Reply • Share ›

Guest • 5 months ago

How can we get the text and pattern as inputs from the user, instead of hard coding them in the program??

• Reply • Share ›

Guest ➚ Guest • 3 months ago

Use getline. :)

1 • Reply • Share ›

Guest • 5 months ago

how do i get the text and pattern inputs from the user, instead of hard coding them in the program??

• Reply • Share ›

Ashish Maheshwari ➚ Guest • 3 months ago

Note: this is for C++.
if u want to take input from a file instead of hard coding, make a file named input.txt in the same folder as the .cpp file and include this line of code in your main() function.

freopen("input.txt","r",stdin);

What it does is, it redirects your input stream (stdin) to the file you provided. Hope this helps :)

1 • Reply • Share ›

amateur • 6 months ago

why is the first case of all characters of text and pattern same a worst case?

• Reply • Share ›

Saksham Gupta ➚ amateur • 6 months ago

Because the inner loop (j) will never break since all the characters are matching. Hence maximum comparisons will me made. Hence, worst case.

2 • Reply • Share ›

SmiH • 8 months ago

I applied the same approach but in more efficient way. So that the complexity has become O(n). Only one loop will be used instead of two.

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
int main()
{
char *str="dimanshumansfjdhfj";
char *str1="mans";
int len1,len2,i=0,j=0;
len1=strlen(str);
len2=strlen(str1);
while(i<len1) {="" if(str[i]="=str1[j])" {="" i++;="" j++;="" if(j="=len2-1)" printf("match="" found="" on="" index="" %d\n",i-j);="" }="" else="" {="" i="i-j;" j="0;" i++;="" }="" }="" return="" 0;="" }="" output:="" match="" found="" at="" index="" 2="" match="" found="" at="" index="" 8.="">
```

• Reply • Share ›

DS+Algo=Placement SmiH • 3 months ago

Though you have used a single loop but due to the statement
i=i-j;
Variable i moves forward as well as backward
so your complexity is not O(n)
If u still didn't understand, consider this loop and tell me its complexity:

```
while(n--)
{
n++;
}
```

Though it is a single loop, so at first, you will think its complexity to be O(n) but n is moving both forward and backward and this loop will never end.

• Reply • Share ›

Martin • 9 months ago

```
public static void findPatterns(char[] str, char[] pattern) {

int i = 0;

int index = 0;

while (i < str.length) {

if (str[i] == pattern[index]) {

index++;

} else {

i =(index);
```

```
i-=(index);

index = 0;

}

if (index == pattern length) {
```

see more

• Reply • Share ›

Guest • 9 months ago

for that we need to traverse the main text to (length_of_text-length_of_pat+1)...
thanks.

1 • Reply • Share ›

Guest • 9 months ago

this code won't work if the pattern is matching at last also ....

Ex- txt="AABAABAAABAABA"

pat="AABA"

1 • Reply • Share ›

rahul • a year ago

can anyone please tell me how can i implement this algo on cuda(Parallel programming)?

• Reply • Share ›

mb1994 ➦ rahul • 10 months ago

Simple. each thread receives its 'index' for txt[], from where it'll start comparing. So for all i from 0 to m-1 in txt[], each i-comparison will be done by thread i.

2 • Reply • Share ›

nani • a year ago

what could be the average case complexity of this algorithm

• Reply • Share ›

pappu • 2 years ago

thanks for the code...

one doubt:: In the "WHAT IS THE WORST CASE?" section, how is the first example a worst case?

• Reply • Share ›

Dynamite ➦ pappu • a year ago

In the first example, you are doing comparison of (n-m+1) substrings in the text with the

pattern. Each comparison continues for m characters in the pattern, hence it is the worst case. Consider the case when the pattern was BBBBB instead of AAAAA, in that case you would have rejected all of n-m+1 subtrings in the text in the very first comparison with the pattern, as compared to m comparisons in case of AAAAA. Hence the first example is worst case

1 • Reply • Share ›

Ironman Dynamite • 10 months ago

Hence there would have been n comparisons => best case => O(n).

• Reply • Share ›

manshi • 2 years ago

The given code has a flaw:
Try to find the pattern "ch" in string "aaaaaaaach"

Output: pattern doesn't exist

Correction:

```
/* instead of N-M, allow the loop till end of string */
for (int i = 0; i <= N ; i++)
{
int j;

/* For current index i, check for pattern match */
for (j = 0; j < M; j++)
{
if (txt[i+j] != pat[j])
break;
}
if (j == M) // if pat[0...M-1] = txt[i, i+1, ...i+M-1]
{
printf("Pattern found at index %d \n", i);
}
}
```

Please correct me if I am wrong

• Reply • Share ›

Chirag Patel manshi • 2 years ago

it works k!!The original explained program works k with yr input!!

• Reply • Share ›

Shiwakant Bharti manshi • 2 years ago

My adaptation of the given code in Java actually works. Please check your logic. Consider this scenario:

i = N, j= 1, txt[i+j] will actually lead to ArrayIndexOutOfBoundsException in Java and uncertain behavior in C/C++.

• Reply • Share ›

gautam • 2 years ago

[sourcecode language="JAVA"]

```
public HashSet<Integer> naivePatternSearch(String pattern, String string) {
HashSet<Integer> index = new HashSet<Integer>();
for (int j = 0; j < string.length(); j++) {
for (int i = 0; i < pattern.length(); i++) {
if (pattern.charAt(i) == string.charAt(j+i)) {
if(i==pattern.length()-1){
index.add(j);
}
} else {
break;
}
}
}
return index;
}
```

• Reply • Share ›

abhishek08aug • 2 years ago

```
#include<stdio.h>
#include<string.h>

void search_pattern(char * str, char * pattern) {
  int str_len=strlen(str);
  int pattern_len=strlen(pattern);
  int i, j;
  for(i=0; i<str_len-pattern_len; i++) {
    for(j=0; j<pattern_len; j++) {
      if(*(str+i+j)!=*(pattern+j)) {
        break;
      }
    }
    if(j==pattern_len) {
      printf("Pattern found at index: %d\n", i);
    }
  }
}
```

see more

1 • Reply • Share ›

meap4aa • 2 years ago

A similar Approach through Recursion:

```
 #include <stdio.h>
#include <string.h>

int my_cmp(char* a,char* b,int i,int n);

int my_cmp(char* a,char* b,int i,int n)
{
                if(*(a+i)==*(b+(strlen(b)-n)))
                {
                        //printf("\nChecking i = %d ",i);
                        //printf("\nNow n = %d ",n);
                if(n==1)
                return 1;
                my_cmp(a,b,++i,--n);
                }
                else
```

see more

3 • Reply • Share ›

alien • 2 years ago

1 more approach could be as below:

```
bool isSubstring(char* src, char* pattern) {
int i=0,j=0, flag=0;
int lenp, lens;
for(i=0;*src+!='';i++);
lens = --i;
for(i=0;*pattern+i!='';i++);
lenp = --i;
i=0;

while((*(src+i)) ! = '')
{

if(*(src+i) == *(pattern+j))
{
j++;
```

```
flag++;
if( flag == lenp)
```

see more

• Reply • Share ›

Yatendra Goel • 2 years ago

The above algo is very similar to previous "Naive Pattern Search" algo but as this one uses 'while' loop while the previous one uses 'for' loop, so novice programmers might have to spend few extra minutes to realize that this is the same algo as previous EXCEPT FEW LINES.

So I have written the above algo again using 'for' loop which is same as previous algo (previous "Naive Pattern Search" algo) except two lines (added comment on those two lines) so that it's very easy to see the difference between two.

```
private void printPatternIndices(char[] text, char[] pattern) {

        for (int i = 0; i < text.length - pattern.length + 1; i++) {

            int j;
            for (j = 0; j < pattern.length; j++) {
                if (text[i + j] != pattern[j]) {
                    break;
                }
            }
```

see more

• Reply • Share ›

raman • 4 years ago

@geeksforgeeks plz post KMP, rabin karp string searching algorithm

ASAP , i am looking forward ..Plese Keep in Posting Such

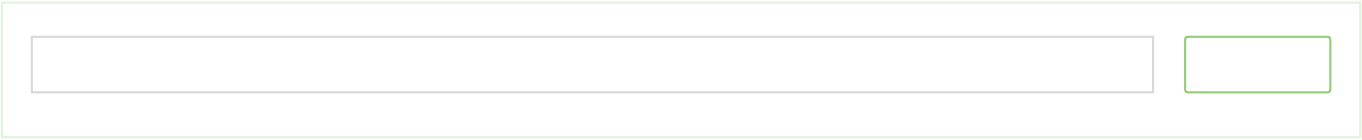
• Reply • Share ›

Vinay ➚ raman • a year ago

1,1,2,3,4,5,6,2,2,2,5,3,3,2,2,1,5,5,5,5,4,4,4,1,2,2,2,2,6,6,2,2,1,1,2,2

Can anyone help me to find out how many times a sequence number (like twice 2,2 or thrice 2,2,2 or four times 2,2,2,2) are repeated in any programming language?

• Reply • Share ›



- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 2 hours ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 2 hours ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 2 hours ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 3 hours ago