Print list items containing all characters of a given word

There is a list of items. Given a specific word, e.g., “sun”, print out all the items in list which contain all the characters of “sum”

For example if the given word is “sun” and the items are “sunday”, “geeksforgeeks”, “utensils”, “”just” and “sss”, then the program should print “sunday” and “utensils”.

Algorithm: Thanks to geek4u for suggesting this algorithm.

```
1) Initialize a binary map:
        map[256] = {0, 0, ..}
2) Set values in map[] for the given word "sun"
        map['s'] = 1,  map['u'] = 1,  map['n'] = 1
3) Store length of the word "sun":
        len = 3 for "sun"
4) Pick words (or items)one by one from the list
    a) set count = 0;
    b) For each character ch of the picked word
         if(map['ch'] is set)
            increment count and unset map['ch']
    c) If count becomes equal to len (3 for "sun"),
           print the currently picked word.
    d) Set values in map[] for next list item
          map['s'] = 1,  map['u'] = 1,  map['n'] = 1
```

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

Google™ Custom Search

GeeksforGeeks

52,731 people like GeeksforGeeks.

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

```
# define NO_OF_CHARS 256

/* prints list items having all caharacters of word */
void print(char *list[], char *word, int list_size)
{
  /*Since calloc is used, map[] is initialized as 0 */
  int *map = (int *)calloc(sizeof(int), NO_OF_CHARS);
  int i, j, count, word_size;

  /*Set the values in map */
  for (i = 0; *(word+i); i++)
      map[*(word + i)] = 1;

  /* Get the length of given word */
  word_size = strlen(word);

  /* Check each item of list if has all characters
     of word*/
  for (i = 0; i < list_size; i++)
  {
    for(j = 0, count = 0; *(list[i] + j); j++)
    {
      if(map[*(list[i] + j)])
      {
         count++;

          /* unset the bit so that strings like
            sss not printed*/
         map[*(list[i] + j)] = 0;
      }
    }
    if(count == word_size)
      printf("\n %s", list[i]);

    /*Set the values in map for next item*/
    for (j = 0; *(word+j); j++)
      map[*(word + j)] = 1;
  }
}

/* Driver program to test to pront printDups*/
int main()
{
  char str[] = "sun";
  char *list[] = {"geeksforgeeks", "unsorted", "sunday", "just", "sss"
  print(list, str, 5);
  getchar();
```

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

```
  return 0;
}
```

Asked by Joey

Time Complexity: O(n + m) where n is total number of characters in the list of items. And m = (number of items in list) * (number of characters in the given word)

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem

Related Tpoics:

- Printing Longest Common Subsequence
- Suffix Array | Set 2 (nLogn Algorithm)
- Rearrange a string so that all same characters become d distance away
- Recursively remove all adjacent duplicates
- Find the first non-repeating character from a stream of characters
- Dynamic Programming | Set 33 (Find if a string is interleaved of two other strings)
- Remove “b” and “ac” from a given string

- Dynamic Programming | Set 29 (Longest Common Substring)

2 Tweet 0 0

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

36 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

Abhi • 2 days ago

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

void PrintList(char* list[],int n,char* str)

{

int i=0;

int len=strlen(str);

int len2;

int x,y,count=0;

while(i<n) {="" len2="strlen(list[i]);" for(x="0;x&lt;len2;x++)" {="" for(y="0;y&lt;le
```

695 Subscribe

Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...

Backtracking | Set 7 (Sudoku) · 20 minutes ago

RVM Can someone please elaborate this Qs from above...

Flipkart Interview | Set 6 · 40 minutes ago

Vishal Gupta I talked about as an Interviewer in general,...

Software Engineering Lab, Samsung Interview | Set 2 · 40 minutes ago

@meya Working solution for question 2 of 4f2f round....

Amazon Interview | Set 53 (For SDE-1) · 1 hour ago

sandeep void rearrange(struct node *head) {...

Given a linked list, reverse alternate nodes and append at the end · 2 hours ago

Neha I think that is what it should return as, in...

Find depth of the deepest odd level leaf node · 2 hours ago

```
printf("\n%s\n",list[i]);
```

see more

^ | ⌄ • Reply • Share ›

Interview Helper • 3 months ago

```
#include<iostream>
#include<string>
#include<cstring>

using namespace std;

#define FORF(i,a,b) for(int i = a; i != b; i++)
#define FORB(i,b,a) for(int i = b; i != a; i--)

#define AL 26

void print(char *list[], char *word, int N){
int *mp = new int[AL];
int len = strlen(word);
FORF(i,0,len) mp[ word[i] - 'a' ] = N;

FORB(i,N-1,-1){
int L = strlen(list[i]);
int count = 0;
```

see more

^ | ⌄ • Reply • Share ›

Interview Helper ➦ Interview Helper • 3 months ago

U don't need re-initialize in this case;

^ | ⌄ • Reply • Share ›

Amit Kumar • 4 months ago

A good idea would be to set the map[256] = {0, 0, ..} value to -1 for the first stri

For the second string2 increment the value (max till 1).
After doing so , check in the map[256] for any - ve value. If there is any then th

• Reply • Share ›

abi • 4 months ago

@geeksforgeeks:wat about if duplicates in both list and given word???

• Reply • Share ›

devesh • 5 months ago

It will fail with word = "suun" even "unsorted", "sunday" contains all the chars c

1 • Reply • Share ›

Guest • 6 months ago

Java implementation without map-
Cosidering 256 chars extended ASCII string
Not as much space efficient as map(but 512 chars in memory at a time shou
but time efficient than map(as map takes some computation to generate hasl
Please provide input for the case it fails--

```
public class AllCharsPresent {

/**
* @param args
*/
public static void main(String[] args) {
// TODO Auto-generated method stub
AllCharsPresent a= new AllCharsPresent();

int i[] =a.makeArray("GfG");
```

int j[]=a.makeArray("GeeksForGeeks");

System.out.println(a.compareArray(i, j));

see more

^ | ˅ • Reply • Share ›

karthik • 6 months ago

it should be a counting array instead of a binary map ... then it works even for
contains duplicates.

^ | ˅ • Reply • Share ›

piyush kumar • 7 months ago

```
#include<stdio.h>

int found(int a[])

{

    int i=0,flag=1;

    for(;i<256;i++)

    {

        if(a[i]>0)

        {

            flag=0;
```

Reply • Share ›

Karan • 9 months ago

```
 /* # include <stdio.h>
# include <stdlib.h>
# include <string.h>
# define NO_OF_CHARS 256

print(char* list[],char* str,int len)
{
    int bin_hash1[NO_OF_CHARS]={0};
    int bin_hash2[NO_OF_CHARS]={0};
    int i = 0,j;
    int found = 0;

    while(*(str + i))
    {
        bin_hash1[*(str + i)]++;
        i++;
    }
```

see more

Reply • Share ›

keyur • 10 months ago

int *map = (int *)calloc(sizeof(int), NO_OF_CHARS);

You did not free this memory. it is memory leak.

```
 /* Paste your code here (You may delete these lines if not writing c
```

^ | ⌄ • Reply • Share ›

anon_user • 11 months ago

Here is code for the same way you have said

```
void checkList(char *listt[],char* word,int sizee)
{
    int i,j,k,flag;
    for(j=0;j<sizee;j++)
    {
        int hash_arr[256] = {0};
        i=0;
        while(*(listt[j]+i)!='&#092&#048')
        {

            if(!hash_arr[*(listt[j]+i)])
               {
                   hash_arr[*(listt[j]+i)]++;
               }
```

see more

^ | ⌄ • Reply • Share ›

neo • a year ago

@geeksforgeeks

if the word contains duplicates like ssun then i think the algo will not work ,pls

```
/* Paste your code here (You may delete these lines if not writing co
```

^ | ⌄ • Reply • Share ›

Sreenivas Doosa ➦ neo • a year ago

@neo

That`s the reason the matched character in map is being unset.

see the following line in the code
map[*(list[i] + j)] = 0;

^ | ⌄ • Reply • Share ›

shruti ➦ Sreenivas Doosa • 11 months ago

Hi geeksforgeeks,

The above mentioned algorithm doesnot work if the string to be
characters.Eg:
word is "ssun" and "sunday" is changed to "ssunday". To reme
frequency of the letters of the word (i.e ssun) in the map and de
it. hence we should not use a binary map. pls correct me if i am

```
/* Paste your code here (You may delete these lines if r
```

2 ^ | ⌄ • Reply • Share ›

Arunkumar Somalinga • a year ago

Here is a very simple implementation of this program using strspn()...this prog
#include<stdio.h>
#include<string.h>
int main()
{

char *str="sun";.

```
char *str2[]={"utensils","geeksforgeeks","sunday","just","sss"};.

int i=0;.

for(i=0;i<5;i++).

{.

if(strspn(str, str2[i])==strlen(str)).

printf("n %s", str2[i]);.

}.

}
```

˄ | ˅ • Reply • Share ›

arunkumar267 • a year ago

this could be very easily done using a library function strspn()...here is a very s

```
#include<stdio.h>
#include<string.h>
int main()
{
    char *str="sun";
    char *str2[]={"utensils","geeksforgeeks","sunday","just","sss"};
    int i=0;
    for(i=0;i<5;i++)
    {
        if(strspn(str,str2[i])==strlen(str))
        printf("\n %s",str2[i]);
    }
}
```

⌃ | ⌄ • Reply • Share ›

Amit • a year ago

```
/* Paste your code here (You may delete these lines if not writing co
#include<stdio.h>
#include<stdlib.h>
#define NO_OF_CHARS 256
void print(char **a,char *b,int len){
     int count[NO_OF_CHARS],temp[NO_OF_CHARS];
     char ch = b[0];
     int i=1,j=0;
     for(j=0;j<256;j++){
                      count[j]=0;
     }
     while(ch != '&#092&#048'){
           count[ch]++ ;
           ch = b[i];
           i++;
     }
     for(j=0;j<256;j++){
```

see more

⌃ | ⌄ • Reply • Share ›

sush • a year ago

gives wrong output when word is "ssun" and "sunday" is changed to "ssunday

```
void print(char *list[], char *word, int l)
{
  int i,*count=(int*)malloc(256*sizeof(int));
  for(i=0;i<l;++i)
  {
        memset(count,0,256*sizeof(int));
        int j,occur=0;
        for(j=0;word[j];++j)
        ++count[word[j]];
        for(j=0;list[i][j];++j)
        if(count[list[i][j]])
        {
                ++occur;
                --count[list[i][j]];
        }
        if(occur==strlen(word))
        printf("%s\n",list[i]);
  }
}
```

^ | ⌄ • Reply • Share ›

Ashish • 2 years ago

This problem can also be solve using bits.
step 1:
grab first item from list.
set each bit corresponding to particular char
once all bits are set
take each character from word that need to be compared for ex 's' from "sum'

space complexity reduced from o(n) to o(1).

^ | ⌄ • Reply • Share ›

Sa • 2 years ago

```
void printItems(char *list[], char *word, int list_size)
{
        map<char, int> countmap;
        map<char,int>::iterator it;
        int i,j, count, flag = 1, len;

        for(i=0;i<strlen(word);i++)
        {
                countmap[word[i]] += 1;
        }

        for(i = 0; i < list_size ;i++)
        {

                for(j=0; *(list[i] +j);j++)
                {
                        if(countmap.count(*(list[i]+j)))
```

see more

^ | ⌄ • Reply • Share ›

S • 2 years ago

```
void printItems(char *list[], char *word, int list_size)
{
        map<char, int> countmap;
        int i,j, count, flag = 1, len;

        for(i=0;i<strlen(word);i++)
        {
                countmap[word[i]] += 1;
        }

        for(i = 0; i < list_size ;i++)
        {

                for(j=0; *(list[i] +j);j++)
                {
                        if(countmap.count(*(list[i]+j)))
```

see more

Reply • Share ›

vkjk89 • 2 years ago

Hi,

Nice post.

Can somebody explain me the time complexity analysis for this ?

How come its O(n+m) ?

I am little weaker in this aspect.Any help would be greatly appreciated.

Also can you give me some URL where i can get better understanding of time

Thanks in advance.

Reply • Share ›

user123 ➦ vkjk89 • 2 years ago

you can guess the time complexity(in most of the cases)just by seeing

here we have...

(no of characters in each list item + length of the word)*no of list items

=no of characters in all the items in the list + length of the word*no of li
=n + m
hence O(n+m)
hope i'm correct...

⌃ | ⌄ • Reply • Share ›

Sunny • 3 years ago

Java solution
use the array of arraylist.
then use the containsAll method for individual element of the array for the giver

⌃ | ⌄ • Reply • Share ›

vaibhav • 3 years ago

what about cases like when str = {'ee'} and list has an element {'geekforgeeks
character in the str is not covered in the algo.

⌃ | ⌄ • Reply • Share ›

jalajb2k7 ➦ vaibhav • 2 years ago

map[e] will be 2 in your case and then above algo will do

```
/* Paste your code here (You may delete these lines if not wri
```

⌃ | ⌄ • Reply • Share ›

Sunil • 3 years ago

I think this is pretty simple way to do however its not efficient.

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

```
int main()
{
        char *list[]={"geeksforgeeks","sunday","utensils"};
        char *word="z";
        int flag,found=0;
        int i,j,k;
        for(i=0;i<3;i++)
        {
                char *temp=list[i];
                found=1;
                for(j=0;word[j]!=NULL;j++)
                {
```

see more

^ | ˅ • Reply • Share ›

sutendra mirajkar • 3 years ago

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



int main()
{
        char str[10],*name[5],tempstr[20];
        int ascstr[10],temp[26];

        printf("ENTER THE STRING:\n");
        scanf("%s",str);

        int i;
        for(i=0;i<strlen(str);i++)
        {
```

see more

Reply • Share ›

mani • 3 years ago

as pointed out by Neil .Since in this we are just counting the set character , if s
times then also the given code is going to print which obviously is WRONG ..!

Reply • Share ›

Anil mani • 3 years ago

```
void print(char *word, char * list[], int list_size)
{
        char *current;
        int wordsize,map[256],i,j,count;

        wordsize = strlen(word);



        for(i=0;i<255;i++)
                map[i]=0;
        for(i=0;i<wordsize;i++)
                map[word[i]]+=1;



        for(i=0;i<list_size;i++)
        {
                count =0;
```

see more

^ | ⌄ • Reply • Share ›

saurabh • 4 years ago

```
[sourcecode language="java"]
private static boolean checkIfQueryIsThere(String inputString, String query) {

int [] inMap = new int[256];
Arrays.fill(inMap, 0);

char [] inCharArr = inputString.toCharArray();
//O(n) id string length is n
for(int i = inCharArr.length-1; i>=0 ; i--)
{
if(inMap[inCharArr[i]] == 0)
inMap[inCharArr[i]] = 1;
}

for(int j=query.length()-1; j>=0 ; j--)
{
if(inMap[query.charAt(j)] == 0)
return false;
}

return true;
}
```

^ | ⌄ • Reply • Share ›

Nell • 4 years ago

I have a small variant of the above question -

How do you find smallest substring containing all characters of a given word e

For Example:

Input string1: "this is a test string"
Input string2: "tist"
Output string: "t stri"

˄ | ˅ • Reply • Share ›

GeeksforGeeks • 4 years ago

@Neil: Thanks for pointing this out. We have made changes to handle this cas

˄ | ˅ • Reply • Share ›

Neil ➦ GeeksforGeeks • 4 years ago

You are welcome.

Actually, if "ss" is specific word, should it considered as one "s" or two
If one "s", this method is ok. But for two "s", it does not work.

My suggestion is to store the number of character in map, not just 0 or
For example, we can do this way for the word

```
for (i = 0; *(word+i); i++)
map[*(word + i)]++; // map['s'] = 2 for "ss"
```

go through list[i], we use

```
for(j = 0; *(list[i] + j); j++)
map[*(list[i] + j)]--;
```

then check if the corresponding map is less than or equal to zero. If ye

Of course, we need to reset the map at each item

˄ | ˅ • Reply • Share ›

Neil • 4 years ago

This one is not really correct.
Suppose given word "ssss".

It will be printed by the above algorithm.