GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About
Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Run Length Encoding

Given an input string, write a function that returns the Run Length Encoded string for the input string.

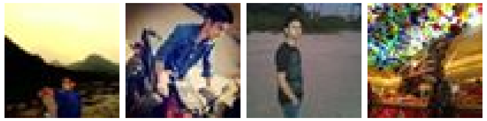
For example, if the input string is “wwwwaaadexxxxxx”, then the function should return “w4a3d1e1x6″.

Algorithm:
a) Pick the first character from source string.
b) Append the picked character to the destination string.
c) Count the number of subsequent occurrences of the picked character and append the count to destination string.
d) Pick the next character and repeat steps b) c) and d) if end of string is NOT reached.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define MAX_RLEN 50

/* Returns the Run Length Encoded string for the
   source string src */
char *encode(char *src)
{
  int rLen;
  char count[MAX_RLEN];
  int len = strlen(src);

  /* If all characters in the source string are different,
    then size of destination string would be twice of input string.
    For example if the src is "abcd", then dest would be "a1b1c1d1"
    For other inputs, size would be less than twice.  */
```



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

```
  char *dest = (char *)malloc(sizeof(char)*(len*2 + 1));

  int i, j = 0, k;

  /* traverse the input string one by one */
  for(i = 0; i < len; i++)
  {

    /* Copy the first occurrence of the new character */
    dest[j++] = src[i];

    /* Count the number of occurrences of the new character */
    rLen = 1;
    while(i + 1 < len && src[i] == src[i+1])
    {
      rLen++;
      i++;
    }

    /* Store rLen in a character array count[] */
    sprintf(count, "%d", rLen);

    /* Copy the count[] to destination */
    for(k = 0; *(count+k); k++, j++)
    {
      dest[j] = count[k];
    }
  }

  /*terminate the destination string */
  dest[j] = '\0';
  return dest;
}

/*driver program to test above function */
int main()
{
  char str[] = "geeksforgeeks";
  char *res = encode(str);
  printf("%s", res);
  getchar();
}
```

Time Complexity: O(n)

References:

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

http://en.wikipedia.org/wiki/Run-length_encoding

Please write comments if you find the above code/algorithm incorrect, or find better 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)

12 Tweet 0 0

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

74 Comments **GeeksforGeeks**

695 Subscribe

Sort by Newest

Join the discussion…

BalaKrishna Reddy · 2 months ago

```
import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class RunLengthEncoding {

public static void main(String[] args) {

String s = "abcccccddegffff";

Pattern pattern = Pattern.compile("(\\w)\\1*");

Matcher matcher = pattern.matcher(s);

StringBuilder sb = new StringBuilder();

String subString = null;

while(matcher.find()){

subString = matcher.group();
```

see more

Reply · Share ›

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

ram93 • 2 months ago

```
#include<stdio.h>

#include<string.h>

int main()

{

int i,j,k,l;

char str[30];

gets(str);

l=strlen(str);

for(i=0;i<l;i++) {="" j="0;" for(k="i;k&lt;l" ;="" k++)="" {="" if(str[k]="=" str[i])="" j+
printf("%c%d",str[i],j);="" i="k-1;" }="" getch();="" return="" 0;="" }="">
```

• Reply • Share ›

Shivji Kumar Jha • 4 months ago

Why isnt dest being freed anywhere?

• Reply • Share ›

tczf1128 • 5 months ago

should be O(N^2)

• Reply • Share ›

Abhay ➦ tczf1128 • 3 months ago

it is (N) as you can see there are two nested loop bt the len value is inc
increment upto lenght only so it is clearly o(N)...!!!!

• Reply • Share ›

sp00ky • 5 months ago

```
public static String encode(String str){

String newstr="";

char alpha=str.charAt(0);

int count=1;

for(int i=1;i<str.length();i++){ if(str.charat(i)="=alpha){" count++;="" }="" else{='
newstr+="(alpha+Integer.toString(count));" alpha="str.charAt(i);" count="1;" }=
newstr+="(alpha+Integer.toString(count));" }="" }="" return="" newstr;="" }="">
```

^ | ⌄ • Reply • Share ›

chandu • 5 months ago

```
#include<stdio.h>

#include<string.h>

void main()

{

char a[]="wwwwaaadexxxxxx";

char s=a[0];

int n=sizeof(a)/sizeof(a[0]);

int count=0,i;

for(i=0;i<n-1;i++) {="" if(a[i]="=a[0])" {="" count++;="" if="" (a[i]!="a[i+1])" {="" p
a[0]="a[i+1];" count="0;" }="" }="" }="" }="">
```

^ | ⌄ • Reply • Share ›

coder • 6 months ago

@GeeksforGeeks
what should be complexity of this code??

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

char str1[]="wwwwaaadexxxxxx";

char str2[20];

int i=0;

int j=0;
```

see more

⌃ | ⌄ • Reply • Share ›

coder • 6 months ago

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()
```

```
{

char str1[]="wwaaabb";

char str2[20];

int i=0;

int j=0;

int count=1;

puts("Your Input String is:");
```

see more

Reply • Share ›

coder • 6 months ago

@GeeksforGeeks:how is that its complexity is O(n) as per my observation it s nesting loops

Reply • Share ›

vibhu • 7 months ago

```
/*most simplified code*/

#include<cstdio>
#include<iostream>
#include<string.h>
#include<cstdlib>
#include<vector>
#include<cmath>
using namespace std;
void runlengthencoding(string& me)
{
int i;
int i;
string ptr;
int count;
for(i=0;i<me.size();i++) {="" count="1;" ptr+="me[i];" while(me[i+1]="=me[i]&ar
count++;="" i++;="" }="" int="" k="0;" int="" num="count;" while(num="">10)
{
num=num/10;
```

see more

⌃ | ⌄ • Reply • Share ›

asunel • 7 months ago

@GeeksforGeeks: Run length encoding using only source string!! But, there a
1) If it is allowed that if a character occurs only once, then its not necessary to

2) First REPEATING character of the source string occurs at least thrice. (Not
but I am not getting how to do it :( )

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

char *encode(char *src)
{
  int rLen, i, j = 0, k;
  int len = strlen(src);

  for(i = 0; i < len; i++)
  {
    src[j++] = src[i];
    rLen = 1;
```

see more

1 ⌃ | ⌄ • Reply • Share ›

1 · Reply · Share ›

iamcoded · 7 months ago

what does sprintf() do?

Reply · Share ›

prabhat · 8 months ago

Hope my had been this much simple but it is having some white spaces also i
Like IF i/p is AABB C C C DDDDDE => o/p A2B2 C C C D5E
spent last 8 hours trying to figure out the algo behind it .

Reply · Share ›

shashi kumar · 8 months ago

```
public static void main(String[] args){
String S="wwwwaaadexxxxxx";
String b=runLen(S);
System.out.print(b);
}

static String runLen(String S){
int l=S.length();
int i=0;
String S1="";
while(i<l-1){ int="" count="1;" while(i<l-1="" &&="" s.charat(i)="=S.charAt(i+1))
s1="S1+S.charAt(i)+count;" i++;="" }="" return="" s1;="" }="">
```

Reply · Share ›

Senthil Kumar · 9 months ago

Krithika Kritz vidu vidu

Reply · Share ›

Krithika Kritz · 9 months ago

dei program ellam ma fb la post pannuva

del program ellam ma fb la post pannuva

^ | ˅ • Reply • Share ›

Senthil Kumar • 9 months ago

```
#include<stdio.h>
#define NO 256.
char str1[];
int main()
{
char str[]="wwweereddccccc";
int i;
getrunlength(str);
for(i=0;str1[i];i++)
{
printf("%c", str1[i]);
}
}
void getrunlength(char str[]).
{
int i, length=0;
int count[NO]={0};
for(i=0;str[i];i++)
```

see more

^ | ˅ • Reply • Share ›

Mishra Navlok • 10 months ago

what if 4 is present 4 times.

^ | ˅ • Reply • Share ›

Vaidyanathan • 11 months ago

If all characters in the source string are different,

then size of destination string would be twice of input string

then size of destination string would be twice of input string.
For example if the src is "abcd", then dest would be "a1b1c1d1"
For other inputs, size would be less than twice. */

This may not be true , because if you have >= ten characters for each charac
i am wrong :)

^ | ⌄ • Reply • Share ›

Ronny ➦ Vaidyanathan • 10 months ago

@Vaidyanathan
If there are more than ten characters, then no doubt we'll use 3 charac
occurrence but you are forgetting the fact that you are replacing more t
characters, so overall you are reducing the length of the string.
Hope it helps.

^ | ⌄ • Reply • Share ›

kaushik • 11 months ago

```
 /kaushik sahu

#include<stdio.h>
#include<conio.h>

int main ()
{
    char *str1 = "wwwwkkshhhhhhhhaa";
    char str2[50];
    int i=0,j=0,count;

    for(i=0;*(str1+i) != '&#092&#048';i++)
    {
        str2[j++] = *(str1+i);
        count = 1;
        while(*(str1 + i+1) == *(str1 + i))
        while( (str1 + i+1) == (str1 + i))
        {
            count++;
```

see more

^ | ⌄ • Reply • Share ›

Krithika Kritz • 11 months ago

```
#include <stdio.h>
#include <stdlib.h>
#include<stdbool.h>
#define RANGE 256.
int main()
{
char str[]="aaaawwwweeee bb";.
runlen(str);
}
void runlen(char *str).
{
int i, k;
int *count =(int*)calloc(RANGE, sizeof(int*));.
for(i=0;*(str+i);i++)
count[*(str+i)]++;
for(k=0;k<RANGE;k++)
if(count[k]>0&& k!=32)
printf("%c->%dn", k,count[k]);
}
```

^ | ⌄ • Reply • Share ›

iftekhar09 • 11 months ago

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
```

```
#include <stdlib.h>
#include <string.h>
void encode(char *str)
{
int n=strlen(str);
int i=0,count=0,k=0,j=0;
char temp;
int *sum=(int*)malloc(sizeof(int)*n);
while(i<n) {="" temp="*(str+i);" while(*(str+i)="=temp)" {="" i++;="" count++;="
*(str+j)="*(str+i-1);" j++;="" count="0;" }="" for(int="" g="0;g&lt;k;g++)" {="" cou
}="" void="" main()="" {="" clrscr();="" char="" s[]="wwwwwaaaabbbbccckka"
```

Reply • Share ›

Iftekhar Ahmed Khan • 11 months ago

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
void encode(char *str).
{
int n=strlen(str);
int i=0, count=0, k=0, j=0;
char temp;
int *sum=(int*)malloc(sizeof(int)*n);
while(i<n)
{
temp=*(str+i);
while(*(str+i)==temp)
{
i++;
count++;
}
```

see more

^ | ⌄ • Reply • Share ›

anmoldhuria • 11 months ago

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
// run length encoding
int main()
{
char str[50] = "waaaaaaaaaadex";
int len=strlen(str);
char stack[50];
char *dest=(char *)malloc(sizeof(char)*2*len+1);
int i=0,j=-1,k=0,top=-1;
printf("%s\n", str);

while(str[i])
{
if(str[i] != str[i+1]) {

dest[k++]=str[i];
int n=i-j;
```

see more

^ | ⌄ • Reply • Share ›

abhishek08aug • a year ago

Will need some more work if there are >=10 occurrences of the same charac

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

```
char * encode_string(char * src) {
  int src_len=strlen(src);
  char * dest=(char *)malloc(sizeof(char)*(src_len*2+1));
  char * current_write_address=dest;
  int i;
  int current_char_len=0;
  char * int_to_char=(char *)malloc(sizeof(char));
  for(i=0; i<src_len; i++){
    *current_write_address=*(src+i);
    current_write_address++;
    current_char_len=1;
    while(i+1<src_len && *(src+i+1)==*(src+i)) {
```

see more

• Reply • Share ›

Hanish • a year ago

We can avoid the use of array count[MAX_RLEN] as

k = sprintf(dest+j , "%d", rLen);
j+=k;

for storing the R_LEN in dest directly

• Reply • Share ›

Star_Trek • a year ago

What if the input is abcab....
is der any way of solving ds in a better and efficient way other than sorting the

• Reply • Share ›

Star_Trek • a year ago

if we have the input as abcda
then output should be a2b1c1d1

is there any way to solve this problem in a better tym complexity other than so
LENGTH ENCODING on it????

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

Reply • Share ›

Gates • a year ago

How to implement this without using extra string??

Reply • Share ›

FirstTime • 2 years ago

```
 #include "stdio.h"
#include "string.h"

char* Encode(char* pStr)
{
        char curr = *pStr;
        char * pEncoded = (char*)malloc((strlen(pStr)*2) +1);
        int count = 1;
        int i = 0;

        while (*pStr)
        {
                if (curr == *(++pStr)) {
                        count++;
                } else {
                        pEncoded[i++] = curr;
                        pEncoded[i++] = (count + '0'); // Get ASCII v
                        count = 1;
```

```
                    count = 1;
```

see more

^ | ˅ • Reply • Share ›

Newbee • 2 years ago

```
char* encode(char*);
int main()
{
char *s="aaaeeebbcdda";
char *t;
t=encode(s);
printf("%s",t);
getch();
return 0;
}

char* encode(char *src)
{
int len=strlen(src);
char*dest=(char*)malloc(sizeof(char)*(len*2+1));
int i=0,j,k=0;
char count;

while(i<len)
```

see more

^ | ˅ • Reply • Share ›

chinnisasi • 2 years ago

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

int main(){
    int i;
    char temp;
    int count;
    char a[] = "aaabbccc";
    int size = sizeof(a)/sizeof(a[0]);
    for(i =0; i < size;i++){
        count = 0;
        temp = a[i];
        cout<<temp;
        while(a[i] == temp){
           count++;
           i = i+ 1;

        }
        i = i -1;
        if(a[i] != '&#92&#48')
           cout<<count;

    }
```

^ | v • Reply • Share ›

chinnisasi ➦ chinnisasi • 2 years ago

i am sry.. for uploading thrice.. there is some problem in uploading..

^ | v • Reply • Share ›

chinnisasi • 2 years ago

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

```
int main(){
    int i;
    char temp;
    int count;
    char a[] = "aaabbccc";
    int size = sizeof(a)/sizeof(a[0]);
    for(i =0; i < 9;i++){
        count = 0;
        temp = a[i];
        cout<<temp;
        while(a[i] == temp){
           count++;
           i = i+ 1;

        }
        i = i -1;
        if(a[i] != '&#92&#48')
           cout<<count;

    }

}
```

Reply • Share ›

chinnisasi ➦ chinnisasi • 2 years ago

please ignore this post

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

Reply • Share ›

• Reply • Share ›

PychoCoder • 2 years ago

Little typo in documentation !!
For example if the src is "abcd", then dest would be "a1b1c1"

while allocating the size of the destination array.

• Reply • Share ›

PsychoCoder PsychoCoder • 2 years ago

it would be

For example if the src is "abcd", then dest would be "a1b1c1d1"

• Reply • Share ›

GeeksforGeeks PsychoCoder • 2 years ago

@PsychoCoder: Thanks for pointing out the typo. We have cor

• Reply • Share ›

tuhin • 2 years ago

```
#include
#include
#include

char *runlength(char *s)
{
int n = strlen(s);
char *dest;
dest = (char*)malloc(sizeof(char)*(2*n));
int count=0;
int i=0,j=0;
char ch;
while(*(s+i)!='')
```

```
{
dest[j] = *(s+i);
j++;
ch = *(s+i);
while(*(s+i)==ch)
```

see more

^ | ⌄ • Reply • Share ›

sudhanshu • 2 years ago

```
char *encode(char *src)
{
int rLen;
int len = strlen(src);

/* If all characters in the source string are different,
then size of destination string would be twice of input string.
For example if the src is "abcd", then dest would be "a1b1c1"
For other inputs, size would be less than twice. */
char *dest = (char *)malloc(sizeof(char)*(len*2 + 1));

int i, j = 0, k;

/* traverse the input string one by one */
for(i = 0; i < len; i++)
{

/* Copy the first occurrence of the new character */
dest[j++] = src[i];

/* Count the number of occurrences of the new character */
rLen = 1;
while(i + 1 < len && src[i] == src[i+1])
{
```

{
rLen++;
i++;
}
dest[j++]=&#0390&#039+rLen;

⌃ | ⌄ • Reply • Share ›

sudhanshu • 2 years ago

in the given solution, no need to take the count[]...
dest[j++]='0'+rLen;
will work..

⌃ | ⌄ • Reply • Share ›

Priyanka • 2 years ago

```
 #include<stdio.h>
#define SIZE 50
void runlen(char *a,char *b)
{
int i,j,count=1;
char c=a[0];
b[0]=c;
for(i=1,j=1;a[i];i++)
{
        if(a[i]==c)
        count++;
        else
        {
        b[j++]=count+'0';
        b[j++]=c=a[i];
        count=1;
        }
}
```

see more

⌃ | ⌄ • Reply • Share ›

Sunny • 3 years ago

Java solution
Use hash map.
pick the character and make it as key.
initially the value is 1 for the key
Run the array if key is already there increment the value by 1
then display the map using the iterator for map.entry

1 ⌃ | ⌄ • Reply • Share ›

vasu • 3 years ago

```
 // not efficient but okay code
//input: aaabbc-->output:a3b2c1
char * run_lenght(const char *str, char *tmp){
        int len=strlen(str);
        int i,k=0;
        char count='1';
        for(i=0;i<len;i++){
                if(str[i]==str[i+1])count++;

                else{
                        tmp[k++]=str[i];
                        tmp[k++]=count;
                    count='1';
                }
        }
        tmp[k]='';
        return tmp;
}
```

see more

1 ⌃ | ⌄ • Reply • Share ›

nameless ➦ vasu • 3 years ago

tmp[k]='';

⌃ | ⌄ • Reply • Share ›

seeker7 • 3 years ago

For the Solution given ,i think the complexity should be O(n^2),as we scan eac
and then count its occurrence in while loop ,so in worst case it is O(n^2)

But if we use if statement instead then its complexity is linear

PLEASE CORRECT ME,IF i am wrong.
THANKS

⌃ | ⌄ • Reply • Share ›

Sandeep ➦ seeker7 • 3 years ago

@seeker7: please take a closer look at the program. The value of i is in
So the statements inside the while loop are executed at most O(n) tim

⌃ | ⌄ • Reply • Share ›

aparna ➦ Sandeep • 3 years ago

The complexity is definitely greater than O(n) !

⌃ | ⌄ • Reply • Share ›

Load more comments

✉ Subscribe   Ⓓ Add Disqus to your site