Find all possible interpretations of an array of digits

Consider a coding system for alphabets to integers where 'a' is represented as 1, 'b' as 2, .. 'z' as 26. Given an array of digits (1 to 9) as input, write a function that prints all valid interpretations of input array.

Examples

```
Input: {1, 1}
Output: ("aa", 'k")
[2 interpretations: aa(1, 1), k(11)]

Input: {1, 2, 1}
Output: ("aba", "au", "la")
[3 interpretations: aba(1,2,1), au(1,21), la(12,1)]

Input: {9, 1, 8}
Output: {"iah", "ir"}
[2 interpretations: iah(9,1,8), ir(9,18)]
```

Please note we cannot change order of array. That means {1,2,1} cannot become {2,1,1}
On first look it looks like a problem of permutation/combination. But on closer look you will figure out that this is an interesting tree problem.
The idea here is string can take at-most two paths:
1. Proces single digit
2. Process two digits
That means we can use binary tree here. Processing with single digit will be left child and two digits will be right child. If value two digits is greater than 26 then our right child will be null as we

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

don't have alphabet for greater than 26.

Let's understand with an example .Array a = {1,2,1}. Below diagram shows that how our tree grows.

```
                        "" {1,2,1}            Codes used in tree
                    /             \               "a" --> 1
                   /               \              "b" --> 2
              "a"{2,1}            "l"{1}          "l" --> 12
             /        \          /     \
            /          \        /       \
         "ab"{1}        "au"    "la"      null
          /    \
         /      \
       "aba"      null
```

Braces {} contain array still pending for processing. Note that with every level, our array size decreases. If you will see carefully, it is not hard to find that tree height is always n (array size)
How to print all strings (interpretations)? Output strings are leaf node of tree. i.e for {1,2,1}, output is {aba au la}.
We can conclude that there are mainly two steps to print all interpretations of given integer array.

Step 1: Create a binary tree with all possible interpretations in leaf nodes.

Step 2: Print all leaf nodes from the binary tree created in step 1.

Following is Java implementation of above algorithm.

```
// A Java program to print all interpretations of an integer array
import java.util.Arrays;

// A Binary Tree node
class Node {

    String dataString;
    Node left;
    Node right;

    Node(String dataString) {
        this.dataString = dataString;
        //Be default left and right child are null.
```

Recommended
Update Windows Drivers

OS	Windows XP, Vista, 7 ,8
Price	Free Trial
Details	Click Here For Details

Download Now

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    }

    public String getDataString() {
        return dataString;
    }
}

public class arrayToAllInterpretations {

    // Method to create a binary tree which stores all interpretations
    // of arr[] in lead nodes
    public static Node createTree(int data, String pString, int[] arr)

        // Invalid input as alphabets maps from 1 to 26
        if (data > 26)
            return null;

        // Parent String + String for this node
        String dataToStr = pString + alphabet[data];

        Node root = new Node(dataToStr);

        // if arr.length is 0 means we are done
        if (arr.length != 0) {
            data = arr[0];

            // new array will be from index 1 to end as we are consumi
            // first index with this node
            int newArr[] = Arrays.copyOfRange(arr, 1, arr.length);

            // left child
            root.left = createTree(data, dataToStr, newArr);

            // right child will be null if size of array is 0 or 1
            if (arr.length > 1) {

                data = arr[0] * 10 + arr[1];

                // new array will be from index 2 to end as we
                // are consuming first two index with this node
                newArr = Arrays.copyOfRange(arr, 2, arr.length);

                root.right = createTree(data, dataToStr, newArr);
            }
        }
        return root;
    }
```


Integrated
Desktop & Mobile Device
Management

- App Management
- Policy Management
- Patch Management
- Software Deployment

Download

ManageEngine
Desktop Central

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    // To print out leaf nodes
    public static void printleaf(Node root) {
        if (root == null)
            return;

        if (root.left == null && root.right == null)
            System.out.print(root.getDataString() + "  ");

        printleaf(root.left);
        printleaf(root.right);
    }

    // The main function that prints all interpretations of array
    static void printAllInterpretations(int[] arr) {

        // Step 1: Create Tree
        Node root = createTree(0, "", arr);

        // Step 2: Print Leaf nodes
        printleaf(root);

        System.out.println();  // Print new line
    }

    // For simplicity I am taking it as string array. Char Array will
    private static final String[] alphabet = {"", "a", "b", "c", "d",
        "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r
        "s", "t", "u", "v", "w", "x", "v", "z"};

    // Driver method to test above methods
    public static void main(String args[]) {

        // aacd(1,1,3,4) amd(1,13,4) kcd(11,3,4)
        // Note : 1,1,34 is not valid as we don't have values correspo
        // to 34 in alphabet
        int[] arr = {1, 1, 3, 4};
        printAllInterpretations(arr);

        // aaa(1,1,1) ak(1,11) ka(11,1)
        int[] arr2 = {1, 1, 1};
        printAllInterpretations(arr2);

        // bf(2,6) z(26)
        int[] arr3 = {2, 6};
        printAllInterpretations(arr3);
```

Subscribe

Recent Comments

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

Backtracking | Set 7 (Sudoku) · 31 minutes ago

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

Flipkart Interview | Set 6 · 51 minutes ago

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

Software Engineering Lab, Samsung Interview | Set 2 · 51 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

AdChoices

► Java to C++

► Java Array

► JavaScript Array

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        // ab(1,2), l(12)
        int[] arr4 = {1, 2};
        printAllInterpretations(arr4);

        // a(1,0} j(10)
        int[] arr5 = {1, 0};
        printAllInterpretations(arr5);

        // "" empty string output as array is empty
        int[] arr6 = {};
        printAllInterpretations(arr6);

        // abba abu ava lba lu
        int[] arr7 = {1, 2, 2, 1};
        printAllInterpretations(arr7);
    }
}
```

Output:

```
aacd  amd  kcd
aaa  ak  ka
bf  z
ab  l
a  j

abba  abu  ava  lba  lu
```

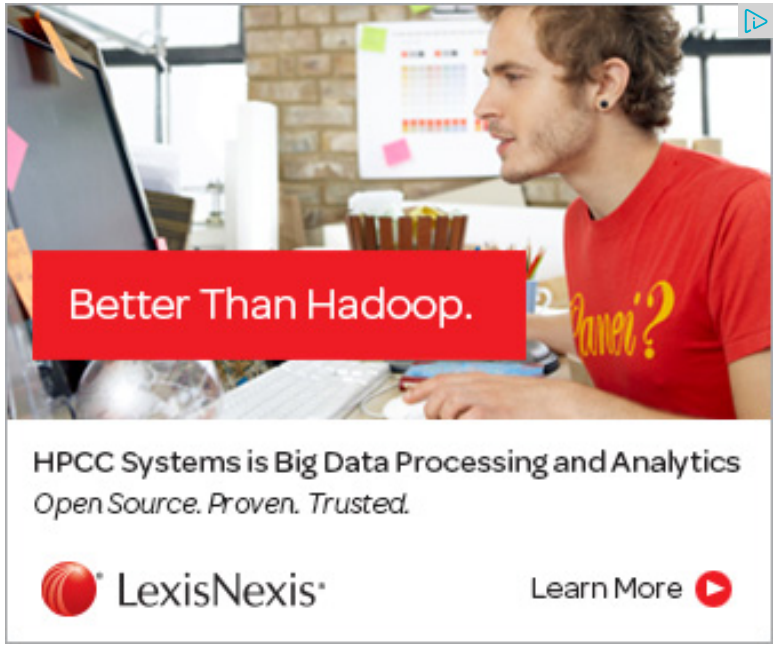
Exercise:

1. What is the time complexity of this solution? [Hint : size of tree + finding leaf nodes]
2. Can we store leaf nodes at the time of tree creation so that no need to run loop again for leaf node fetching?
3. How can we reduce extra space?

This article is compiled by Varun Jain. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

AdChoices
► IR Array
► An Array
► Jquery Array
AdChoices
► C++ Array
► String Array
► C++ Coding

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com



Related Tpoics:

- Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)
- Print Right View of a Binary Tree
- Red-Black Tree | Set 3 (Delete)
- Construct a tree from Inorder and Level order traversals
- Print all nodes at distance k from a given node
- Print a Binary Tree in Vertical Order | Set 1
- Interval Tree
- Check if a given Binary Tree is height balanced like a Red-Black Tree

21 Tweet 3 1

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

43 Comments GeeksforGeeks

Sort by Newest

Join the discussion

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Join the discussion…

AlienOnEarth • 3 days ago

Another non-tree based solution using recursion:

```
#include<stdio.h>

int a[3]={1, 2, 1};

char c[]={'0','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

void func(int i,char result[3],int r)

{

int index;

int temp;

if(i>=3)

{

for(index=0;index<3;index++)
```

see more

• Reply • Share ›

Guest • 5 days ago

```
tree* possible(string word, queue<char> temp)

{

tree *root=newnode(word);

if(temp.empty())
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
{

cout<<root->str<<endl; return="" root;="" }="" int="" a="temp.front()-'0';" temp.
="">left=possible(word+char(96+a),temp);}

if(!temp.empty())

{

a=a*10+temp.front()-'0';

temp.pop();

root->right=possible(word+char(96+a),temp);

}

return root;

}
```

• Reply • Share ›

bhopu • 10 days ago

change in leaf to hold modified string

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct tree{

char *str;
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
struct tree *lchild;

struct tree *rchild;

};

struct tree *getnode(char *str,int n){

struct tree *newnode=(struct tree *)malloc(sizeof(struct tree));
```

see more

• Reply • Share ›

naveenbobbili • 2 months ago

Below is my solution using only recursion.

```
/*
* patterns.cpp
*
* Created on: Feb 16, 2014
* Author: naveen1.b
*/
#include<iostream>
#include<vector>
#include<string>
#include<list>
#include<stdlib.h>

using namespace std;

static string alphabets[] = { " ",
"a", "b", "c", "d", "e", "f", "g", "h", "i",
"j", "k", "l", "m", "n", "o", "p", "q", "r",
```

see more

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

• Reply • Share ›

Shivam • 6 months ago

No need to make tree.

void dfs(int a[],int i,int prev,int n,string s)

{

if(i==n)

{

s=s+char(prev+'a'-1);

cout<<s<<endl; return;="" }="" dfs(a,i+1,a[i],n,s+char(prev+'a'-1));="" if(prev*1
dfs(a,i+1,prev*10+a[i],n,s);="" }="" }="" int="" main()="" {="" int="" t;="" cin="">:

while(t--)

{

int n;

see more

• Reply • Share ›

Vijay Apurva • 6 months ago

we can do simply with out any extra space .....

#include<stdio.h>

char arr[] = {' ','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't

void printc(int x[] , int y , char c[],int n ,int check){

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
if(n == y){
c[check]='\0' ;
printf("%s \n",c);
return ;
}
if(y>n){
c[check] = arr[x[n]];
```

see more

Reply • Share ›

Amit Bgl • 9 months ago

wow code :D

Reply • Share ›

ankur jain • 9 months ago

```
 #include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>

#define input freopen("input.txt","r",stdin)
#define output freopen("out.txt","w",stdout)
//a=a+b-(b=a);
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
using namespace std;
/*
struct tree
{
        int data;
```

see more

1 • Reply • Share ›

Umang Mahajan ➦ ankur jain • a month ago

great!! Just one modification...do you really need to pass 'd' as a parar

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

string alpha[] = {"", "a", "b", "c", "d", "e","f", "g", "h", "i", "j", "k", "l", "m", "n
"x", "v", "z"};

void create(int arr[],int n,string res)
{
if (n==0)
{
cout<<res<<endl; return;="" }="" int="" d="arr[0];" create(arr+1,n-1,res·
{
d=arr[0]*10+arr[1];
if (d < 27)
create (arr+2,n-2,res+alpha[d]);
```

see more

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

• Reply • Share ›

ankur jain • 9 months ago

i have done this only by recursion doesnot make any tree
then why their is need to build tree ?
please correct me..if am wrong (or if by taking tree is better apporach then mi

```
 #include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>

#define input freopen("input.txt","r",stdin)
#define output freopen("out.txt","w",stdout)
//a=a+b-(b=a);


using namespace std;
```

see more

• Reply • Share ›

bateesh • 9 months ago

A very simple solution which assumes that at a particular index we can put a a
2 digits as 26 is max range of alphabets.
Like for 1,2,1
for first index we can have 1 or 12
3 digits combination is not possible.Following is the implementation.

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

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
#include<conio.h>
void permute(int *arr,int s,int e,int idx,char *str);
int main()
{
    int arr[20];
    char str[20];
    int n;
    printf("\n enter the total digits:");
    cin>>n;
    for(int i=0;i<n;i++)
    cin>>arr[i];
```

see more

2 • Reply • Share ›

xxmajia • 10 months ago

Shall we change "Given an array of digits (1 to 9) as input" to
"Given an array of digits (0 to 9) as input"?
if there is no 0, we can not interpret 10 and 20
Anyway, the solution is correct and considering 0 as part of the input

[sourcecode language="JAVA"]
/* Paste your code here (You may delete these lines if not writing code) */
public static ArrayList<ArrayList<Character>> findAll (int[] A) {
ArrayList<ArrayList<Character>> results = new ArrayList<ArrayList<Characte
if (A == null || A.length == 0) {
return results;
}

doFindAll(A, 0, results, new ArrayList<Character>());
return results;
}

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

public static void doFindAll (int[] A, int position, ArrayList<ArrayList<Character>
result) {

see more

• Reply • Share ›

xxmajia • 10 months ago

Shall we change "Given an array of digits (1 to 9) as input" to
"Given an array of digits (0 to 9) as input"?
if there is no 0, we can not interpret 10 and 20
Anyway, the solution is correct and considering 0 as part of the input

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

• Reply • Share ›

MVN Murthy • 10 months ago

//It can be easily solved by iteration. no need of recursion and no need of creat
//Space complexity (1)

[sourcecode language="python"]
x = [int(i) for i in raw_input().split()]
n = len(x)
for i in x:
print chr(96+i),

print
for i in range(n-1):
a=int(str(x[i])+str(x[i+1]))
if a>26: continue

for j in range(i): print chr(x[j]+96),

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

print chr(96+a),

for j in range(i+2,n): print chr(x[j]+96)

• Reply • Share ›

Rahul → MVN Murthy • 10 months ago

can you please explain your logic?

• Reply • Share ›

bateesh • 10 months ago

This can be done easily with recursion and many people has already submitte
complexity for the recursive solution?I dnt think it will have overlapping subprob

• Reply • Share ›

shek8034 → bateesh • 10 months ago

Complexity is O(n) in worst case, if the tree formed is skewed.

• Reply • Share ›

bateesh → shek8034 • 10 months ago

@shek8034.
I think worst case for this would not be 0(n).The complexity for
also be deduced from the tree shown in above diagram.plz con

• Reply • Share ›

shek8034 → bateesh • 10 months ago

@bateesh : Yes you are right. Complexity is O(2^n) sinc

My bad. :P Please ignore my previous comment.

• Reply • Share ›

shek8034 • 10 months ago

No need to build Binary tree Easily done Using recursion

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

No need to build Binary tree.Easily done using recursion.
Because recursion itself makes the binary tree.

A very simple recursive code.
Please comment if you find any problem
Thanks :)

```
 #include<iostream>
#include<string>
#include<stdio.h>
using namespace std;

int arr[]={1,1,3,4};
int n = sizeof(arr)/sizeof(arr[0]);

void interpretation(string str,int idx)
{
        if(idx == n)
        {
```

see more

• Reply • Share ›

illuminati • 10 months ago

It can be done by recursion easily... Below is the source code for it.

```
 #include<iostream>
#include<cstring>
#include<sstream>
using namespace std;
int n;
void dig_combs(int arr[],int k,string ans)
{
     int r;
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    if(k==n){
    cout<<ans<<endl;
    int temp;
    stringstream s (ans);
    while(s>> temp){
    char ch=(char)(temp-1)+'a';
    //cout<<"temp: "<<temp<<endl;
    cout<<ch<<" ";
```

see more

Reply • Share ›

Rahul • 10 months ago

```
 /* #include <iostream>
#include <string>
#include <vector>

using namespace std;

char alphabet[] = {' ', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
                                 'n', 'o', 'p', 'q', 'r', 's', 't',

typedef struct node
{
        string value;
        struct node *left;
        struct node *right;

        node()
        {
                value = " ";
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

see more

• Reply • Share ›

Srinivas Masna • 10 months ago

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

using namespace std;.

void printAllInterpretations(int a[], int s, int e, char res[], int i);.
void printres(char res[], int i);.

char alphabet[] = {" abcdefghijklmnoprstuvwxyz"};.

int main()
{

int a[] = {1, 2, 1};.
char res[3] ="";.
printAllInterpretations(a, 0, 3, res, 0);.
}

void printAllInterpretations(int a[], int s, int e, char res[], int i).
{
```

see more

• Reply • Share ›

EOF • 10 months ago

Another way of thinking:

1) Let toChar(i) denotes the character corresponding to digit i.
2) Let f(i,j) denotes all the interpretations of the array arr[i..j]. The problem can

```
f(i,j) = toChar(arr[i]) UNION f(i+1,j);
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        if(10*arr[i]+arr[i+1] <= 26)
                f(i,j) = f(i,j) UNION tochar(10*arr[i]+arr[i+1]) UNIO
```

For simplicity I didn't do the bound checking for array indexes and base cases
Can be implemented with O(n) time complexity using Dynamic Programming.

• Reply • Share ›

atul • 11 months ago

output for below test case is wrong :-

```
int arr[]={1, 1, 3,0,0};
```

output : aac aac am am kc kc

expected output : aac am kc

• Reply • Share ›

coder → atul • 11 months ago

your input case is wrong since in the question it is clearly given that "ar
If you find something unconvincing then do reply to this post.

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

• Reply • Share ›

atul → coder • 11 months ago

given code itself have similar test case :-

// a(1,0} j(10)
int[] arr5 = {1, 0};
printAllInterpretations(arr5);

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

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

• Reply • Share ›

coder ➦ atul • 11 months ago

kk..actually my code is written as per the question given
there of java so consider the code only when the given a
won't be much change in the code to make it valid for te

• Reply • Share ›

Manzil Roy • 11 months ago

Tuhin Chakrabarty : ya correct, bt the simple old recursion will be too costly. T
programming table) ought to be modified to store/print the partial solutions wh
subproblems.

• Reply • Share ›

Tuhin Chakrabarty • 11 months ago

dynamic programming will give you the count of interpretations . for printing the
the solution as memoization is difficult .a naive recursion can serve the purpos
repeatedly . (exponential complexity)
i guess you are thinking about this
http://www.spoj.com/SPOJ/probl... . this is similar stuff . just it expects the nur
with DP

• Reply • Share ›

Manzil Roy • 11 months ago

Can we solve this by using Dynamic Programming?

• Reply • Share ›

coder • 11 months ago

I think rather than representing the string as physical tree,it is better to use rec

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

[sourcecode language="C++"]
#include<iostream>
using namespace std;
#define MAX 100
char arr[27];
int input[MAX];
int n;
void permut(char *a,int index,int,int);
int main()
{

cout<<"enter no. of elements"<<endl;
cin>>n;
cout<<"enter the elements"<<endl;
for(int i=0;i<n;i++)
cin>>input[i];
for(int i=1;i<=26;i++)

see more

• Reply • Share ›

coder • 11 months ago

I think recursion will work just fine.No need to represent it as a tree physically.I

[sourcecode language="C++"]

• Reply • Share ›

kk.nitrkl • 11 months ago

[sourcecode language="C++"]
#include <iostream>
#include <string>

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

using namespace std;

bool is_valid(int a)
{
return a >= 65 && a <= 90;
}

void interpret(int arr[], string str, int curr, int len)
{
if(curr == len)
{
cout<<str<<endl;
}
else
{

see more

• Reply • Share ›

nitish712 • 11 months ago

```
 #include <stdio.h>
int vals[10];
int vcnt;
int n=8;
int arr[]={1,2,3,1,2,3,2,4};
void compute(int lev, int idx)
{
        if(lev==n)
        {
                check(idx);
                return;
        }
        vals[idx] = arr[lev];
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        compute(lev+1, idx+1);
        if(lev>=n-1)
                return;
        vals[idx] = vals[idx]*10 + arr[lev+1];
        if(vals[idx]>26)
```

see more

Reply • Share ›

AMIT • 11 months ago

print the output without building the tree

```
 #include<stdio.h>
#include<stdlib.h>
void print1(int a[],char b[],int i1,int n,int i2,int flag)
{
    if(i1==n && flag==1)
    return;
    if(i1==n)
    {
        b[i2]='&#92&#48';
        printf("%s\t",b);
    }
    if(flag==0 && a[i1]==0)
    return;
    if(flag==1 && (a[i1-1]>2 ||(a[i1-1]==2 && a[i1]>6)))
    return;
    if(flag==0)
```

see more

Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

ankitjaingc • 11 months ago

What is the best case complexity ??
I could come up with O(n2)=O(n square).
Please suggest better method.

• Reply • Share ›

ankitjaingc • 11 months ago

#include<stdio.h>

int Possible(int *A,int n)
{
int i=0,j=0;
for(i=0;i<n;i++) printf("%c",a[i]+96);="" printf("\n");="" for(i="0;i<n-1;i++)" {="" for(j="0;j<i;j++)" printf("%c",a[j]+96);="" printf("%c",a[i]*10+a[i+1]+96);="" for(printf("%c",a[j]+96);="" printf("\n");="" }="" }="" }="" main()="" {="" int="" a[]="{9, return="" 0;="" }="">

• Reply • Share ›

Lam • 11 months ago

I think we can use recursion, the pros is that it does not require to store the lea is many function calls which maybe inefficient. However, there will be no algor worst case since the output in the worst case will be that number.

```
 #include <stdio.h>
#include <string>
#include <iostream>
using namespace std;

/**
* return a character corresponding to the digit, e.g. 1->a, 2->b, ...,
*/
char digitToChar(int d){
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        const string s="abcdefghijklmnopqrstuvwxyz";
        if(d>=27||d<=0)
                return '#';
        return s[d-1];
}
```

see more

Reply • Share ›

nitin35 • 11 months ago

creating a complete tree is not necessary
following should work just fine i think
[sourcecode language="C++"]
/* Paste your code here (You may delete these lines if not writing code) */
#include<iostream>
#include<vector>
using namespace std;
string alpha[27]={" ","a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r

void process(string s,vector<int> rem)
{
int a,b,c;
if(rem.empty())
cout<<s<<endl;

else
{
//process one length

see more

Reply • Share ›

GeeksforGeeks ➦ nitin35 • 11 months ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@nitin35: Could you please post your code again between sourcecode

• Reply • Share ›

nitin35 ➦ GeeksforGeeks • 11 months ago

```
 #include<iostream>
#include<vector>
using namespace std;
string alpha[27]={" ","a","b","c","d","e","f","g","h","i



void process(string s,vector<int> rem)
{
    int a,b,c;
    if(rem.empty())
        cout<<s<<endl;

    else
    {
        //process one length
            a=rem[0];
            rem.erase(rem.begin());
            process(s+alpha[a],rem);
```

see more

• Reply • Share ›

Dheeraj • 11 months ago

Total count of interpretations can be written as following:

count(arr, n) = n if n = 1 or n = 0

count(arr, n) = count(arr+1, n-1) + count(arr+2, n-2) if n > 2 and arr[0]*10 + ar

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

count(arr, n) = count(arr+1, n-1) otherwise

In ur solution, u have used a Binary Tree to store results of sub-problems beca
subproblems, let me know if my understanding is correct.

• Reply • Share ›

Subscribe Add Disqus to your site

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com