

Topics :

- [Analysis of Algorithms](#)
- [Searching and Sorting](#)
- [Greedy Algorithms](#)
- [Dynamic Programming](#)
- [Pattern Searching](#)
- [Other String Algorithms](#)
- [Backtracking](#)
- [Divide and Conquer](#)
- [Geometric Algorithms](#)
- [Mathematical Algorithms](#)
- [Bit Algorithms](#)
- [Graph Algorithms](#)
- [Randomized Algorithms](#)
- [Quizzes on Algorithms](#)
- [Misc](#)

Analysis of Algorithms:

- [Asymptotic Analysis](#)
- [Worst, Average and Best Cases](#)
- [Asymptotic Notations](#)
- [Analysis of Loops](#)
- [Solving Recurrences](#)
- [Amortized Analysis](#)
- [What does 'Space Complexity' mean?](#)
- [NP-Completeness Introduction](#)
- [A Time Complexity Question](#)
- [Time Complexity of building a heap](#)

[Quiz on Analysis of Algorithms](#)

[Quiz on Recurrences](#)

Searching and Sorting:

- Binary Search
- Selection Sort
- Bubble Sort
- Insertion Sort
- Merge Sort
- Heap Sort
- QuickSort
- Bucket Sort
- ShellSort
- Interpolation search vs Binary search
- Stability in sorting algorithms
- When does the worst case of Quicksort occur?
- Lower bound for comparison based sorting algorithms
- Which sorting algorithm makes minimum number of memory writes?
- Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted
- Merge Sort for Linked Lists
- Sort a nearly sorted (or K sorted) array
- Iterative Quick Sort
- QuickSort on Singly Linked List
- QuickSort on Doubly Linked List
- Find k closest elements to a given value
- Sort n numbers in range from 0 to $n^2 - 1$ in linear time
- A Problem in Many Binary Search Implementations
- Search in an almost sorted array
- Sort an array in wave form
- Why is Binary Search preferred over Ternary Search?
- K'th Smallest/Largest Element in Unsorted Array
- K'th Smallest/Largest Element in Unsorted Array in Expected Linear Time
- K'th Smallest/Largest Element in Unsorted Array in Worst Case Linear Time
- Find the closest pair from two sorted arrays
- Find common elements in three sorted arrays
- Given a sorted array and a number x, find the pair in array whose sum is closest to x
- Count 1's in a sorted binary array
- Binary Insertion Sort
- Insertion Sort for Singly Linked List
- Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?
- Merge Sort for Doubly Linked List

Quiz on Sorting

Quiz on Searching

Greedy Algorithms:

- Activity Selection Problem
- Kruskal's Minimum Spanning Tree Algorithm
- Huffman Coding
- Efficient Huffman Coding for Sorted Input
- Prim's Minimum Spanning Tree Algorithm
- Prim's MST for Adjacency List Representation
- Dijkstra's Shortest Path Algorithm
- Dijkstra's Algorithm for Adjacency List Representation
- Job Sequencing Problem
- Quiz on Greedy Algorithms
- K Centers Problem

Dynamic Programming:

- Overlapping Subproblems Property
- Optimal Substructure Property
- Longest Increasing Subsequence
- Longest Common Subsequence
- Edit Distance
- Min Cost Path
- Coin Change
- Matrix Chain Multiplication
- Binomial Coefficient
- 0-1 Knapsack Problem
- Egg Dropping Puzzle
- Longest Palindromic Subsequence
- Cutting a Rod
- Maximum Sum Increasing Subsequence
- Longest Bitonic Subsequence
- Floyd Warshall Algorithm
- Palindrome Partitioning
- Partition problem
- Word Wrap Problem
- Maximum Length Chain of Pairs
- Variations of LIS
- Box Stacking Problem
- Program for Fibonacci numbers
- Minimum number of jumps to reach end
- Maximum size square sub-matrix with all 1s

- Ugly Numbers
- Largest Sum Contiguous Subarray
- Longest Palindromic Substring
- Bellman–Ford Algorithm for Shortest Paths
- Optimal Binary Search Tree
- Largest Independent Set Problem
- Subset Sum Problem
- Maximum sum rectangle in a 2D matrix
- Count number of binary strings without consecutive 1's
- Boolean Parenthesization Problem
- Count ways to reach the n'th stair
- Minimum Cost Polygon Triangulation
- Mobile Numeric Keypad Problem

See [Dynamic Programming Tag](#) for more problems, [Quiz on Dynamic Programming](#)

Pattern Searching:

- Naive Pattern Searching
- KMP Algorithm
- Rabin-Karp Algorithm
- A Naive Pattern Searching Question
- Finite Automata
- Efficient Construction of Finite Automata
- Boyer Moore Algorithm – Bad Character Heuristic
- Suffix Array
- Anagram Substring Search (Or Search for all permutations)
- Pattern Searching using a Trie of all Suffixes

Other String Algorithms:

- Manacher's Algorithm – Linear Time Longest Palindromic Substring – Part 1, Part 2, Part 3, Part 4
- Longest Even Length Substring such that Sum of First and Second Half is same
- Print all possible strings that can be made by placing spaces

Backtracking:

- Print all permutations of a given string
- The Knight's tour problem
- Rat in a Maze
- N Queen Problem
- Subset Sum
- m Coloring Problem

- Hamiltonian Cycle
- Sudoku
- Tug of War
- Solving Cryptarithmic Puzzles

Divide and Conquer:

- Introduction
- Write your own pow(x, n) to calculate x^n
- Median of two sorted arrays
- Count Inversions
- Closest Pair of Points
- Strassen's Matrix Multiplication

See this for more, Quiz on Divide and Conquer

Geometric Algorithms:

- Closest Pair of Points | $O(n \log n)$ Implementation
- How to check if two given line segments intersect?
- How to check if a given point lies inside or outside a polygon?
- Convex Hull | Set 1 (Jarvis's Algorithm or Wrapping)
- Convex Hull | Set 2 (Graham Scan)
- Given n line segments, find if any two segments intersect
- Check whether a given point lies inside a triangle or not
- How to check if given four points form a square

Mathematical Algorithms:

- Write an Efficient Method to Check if a Number is Multiple of 3
- Efficient way to multiply with 7
- Write a C program to print all permutations of a given string
- Lucky Numbers
- Write a program to add two numbers in base 14
- ~~Babylonian method for square root~~
- ~~Multiply two integers without using multiplication, division and bitwise operators, and no loops~~
- Print all combinations of points that can compose a given number
- Write you own Power without using multiplication(*) and division(/) operators
- Program for Fibonacci numbers
- Average of a stream of numbers
- Count numbers that don't contain 3
- MagicSquare
- Sieve of Eratosthenes

- Find day of the week for a given date
- DFA based division
- Generate integer from 1 to 7 with equal probability
- Given a number, find the next smallest palindrome
- Make a fair coin from a biased coin
- Check divisibility by 7
- Find the largest multiple of 3
- Lexicographic rank of a string
- Print all permutations in sorted (lexicographic) order
- Shuffle a given array
- Space and time efficient Binomial Coefficient
- Reservoir Sampling
- Pascal's Triangle
- Select a random number from stream, with $O(1)$ space
- Find the largest multiple of 2, 3 and 5
- Efficient program to calculate e^x
- Measure one litre using two vessels and infinite water supply
- Efficient program to print all prime factors of a given number
- Print all possible combinations of r elements in a given array of size n
- Random number generator in arbitrary probability distribution fashion
- How to check if a given number is Fibonacci number?
- Russian Peasant Multiplication
- Count all possible groups of size 2 or 3 that have sum as multiple of 3
- Tower of Hanoi
- Horner's Method for Polynomial Evaluation
- Count trailing zeroes in factorial of a number
- Program for n th Catalan Number
- Generate one of 3 numbers according to given probabilities
- Find Excel column name from a given column number
- Find next greater number with same set of digits
- Count Possible Decodings of a given Digit Sequence
- Calculate the angle between hour hand and minute hand
- Count number of binary strings without consecutive 1's
- Find the smallest number whose digits multiply to a given number n
- Draw a circle without floating point arithmetic
- How to check if an instance of 8 puzzle is solvable?
- Birthday Paradox
- Multiply two polynomials
- Count Distinct Non-Negative Integer Pairs (x, y) that Satisfy the Inequality $x*x + y*y < n$
- Count ways to reach the n 'th stair
- Replace all '0' with '5' in an input Integer

- Program to add two polynomials
- Print first k digits of $1/n$ where n is a positive integer
- Given a number as a string, find the number of contiguous subsequences which recursively add up to 9

Bit Algorithms:

- Find the element that appears once
- Detect opposite signs
- Set bits in all numbers from 1 to n
- Swap bits
- Add two numbers
- Smallest of three
- A Boolean Array Puzzle
- Set bits in an (big) array
- Next higher number with same number of set bits
- Optimization Technique (Modulus)
- Add 1 to a number
- Multiply with 3.5
- Turn off the rightmost set bit
- Check for Power of 4
- Absolute value (abs) without branching
- Modulus division by a power-of-2-number
- Minimum or Maximum of two integers
- Rotate bits
- Find the two non-repeating elements in an array
- Number Occurring Odd Number of Times
- Check for Integer Overflow
- Little and Big Endian
- Reverse Bits of a Number
- Count set bits in an integer
- Number of bits to be flipped to convert A to B
- Next Power of 2
- Check if a Number is Multiple of 3
- Find parity
- Multiply with 7
- Find whether a no is power of two
- Position of rightmost set bit
- Binary representation of a given number
- Swap all odd and even bits
- Find position of the only set bit
- Karatsuba algorithm for fast multiplication
- How to swap two numbers without using a temporary variable?

- Check if a number is multiple of 9 using bitwise operators
- Swap two nibbles in a byte
- How to turn off a particular bit in a number?
- Check if binary representation of a number is palindrome

Quiz on Bit Algorithms

Graph Algorithms:

Introduction, DFS and BFS:

- Graph and its representations
- Breadth First Traversal for a Graph
- Depth First Traversal for a Graph
- Applications of Depth First Search
- Detect Cycle in a Directed Graph
- Detect Cycle in a an Undirected Graph
- Detect cycle in an undirected graph
- Longest Path in a Directed Acyclic Graph
- Topological Sorting
- Check whether a given graph is Bipartite or not
- Snake and Ladder Problem
- Biconnected Components
- Check if a given graph is tree or not

Minimum Spanning Tree:

- Prim's Minimum Spanning Tree (MST))
- Applications of Minimum Spanning Tree Problem
- Prim's MST for Adjacency List Representation
- Kruskal's Minimum Spanning Tree Algorithm
- Boruvka's algorithm for Minimum Spanning Tree

Shortest Paths:

- Dijkstra's shortest path algorithm
- Dijkstra's Algorithm for Adjacency List Representation
- Bellman–Ford Algorithm
- Floyd Warshall Algorithm
- Johnson's algorithm for All-pairs shortest paths
- Shortest Path in Directed Acyclic Graph
- Some interesting shortest path questions
- Shortest path with exactly k edges in a directed and weighted graph

Connectivity:

- Find if there is a path between two vertices in a directed graph
- Connectivity in a directed graph
- Articulation Points (or Cut Vertices) in a Graph
- Biconnected graph
- Bridges in a graph
- Eulerian path and circuit
- Fleury's Algorithm for printing Eulerian Path or Circuit
- Strongly Connected Components
- Transitive closure of a graph
- Find the number of islands
- Count all possible walks from a source to a destination with exactly k edges
- Euler Circuit in a Directed Graph
- Biconnected Components
- Tarjan's Algorithm to find Strongly Connected Components

Hard Problems:

- Graph Coloring (Introduction and Applications)
- Greedy Algorithm for Graph Coloring
- Travelling Salesman Problem (Naive and Dynamic Programming)
- Travelling Salesman Problem (Approximate using MST)
- Hamiltonian Cycle
- Vertex Cover Problem (Introduction and Approximate Algorithm)
- K Centers Problem (Greedy Approximate Algorithm)

Maximum Flow:

- Ford-Fulkerson Algorithm for Maximum Flow Problem
- Find maximum number of edge disjoint paths between two vertices
- Find minimum s-t cut in a flow network
- Maximum Bipartite Matching
- Channel Assignment Problem

Misc:

- Find if the strings can be chained to form a circle
- Given a sorted dictionary of an alien language, find order of characters
- Karger's algorithm for Minimum Cut

Quiz on Graph

Quiz on Graph Traversals

Quiz on Graph Shortest Paths

Quiz on Graph Minimum Spanning Tree

Randomized Algorithms:

- Linearity of Expectation
- Expected Number of Trials until Success
- Karger's algorithm for Minimum Cut
- K'th Smallest/Largest Element in Unsorted Array | Set 2 (Expected Linear Time)
- Reservoir Sampling
- Shuffle a given array
- Select a Random Node from a Singly Linked List

Quizzes on Algorithms:

- Analysis of Algorithms
- Sorting
- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Backtracking
- Misc
- NP Complete
- Searching
- Analysis of Algorithms (Recurrences)
- Recursion
- Bit Algorithms
- Graph Traversals
- Graph Shortest Paths
- Graph Minimum Spanning Tree

Misc:

- Commonly Asked Algorithm Interview Questions | Set 1
- Given a matrix of 'O' and 'X', find the largest subsquare surrounded by 'X'
- Nuts & Bolts Problem (Lock & Key problem)
- Flood fill Algorithm – how to implement fill() in paint?
- Given n appointments, find all conflicting appointments
- Check a given sentence for a given set of simple grammar rules
- Find Index of 0 to be replaced with 1 to get longest continuous sequence of 1s in a binary array
- How to check if two given sets are disjoint?
- Minimum Number of Platforms Required for a Railway/Bus Station

- Length of the largest subarray with contiguous elements | Set 1
- Length of the largest subarray with contiguous elements | Set 2
- Print all increasing sequences of length k from first n natural numbers
- Given two strings, find if first string is a subsequence of second
- Snake and Ladder Problem
- Write a function that returns 2 for input 1 and returns 1 for 2
- Connect n ropes with minimum cost
- Find the number of valid parentheses expressions of given length
- Longest Monotonically Increasing Subsequence Size (N log N): Simple implementation
- Generate all binary permutations such that there are more 1's than 0's at every point in all permutations
- Lexicographically minimum string rotation
- Construct an array from its pair-sum array
- Program to evaluate simple expressions
- Check if characters of a given string can be rearranged to form a palindrome
- Print all pairs of anagrams in a given array of strings

Please see [Data Structures and Advanced Data Structures](#) for Graph, Binary Tree, BST and Linked List based algorithms.

We will be adding more categories and posts to this page soon.

You can create a new Algorithm topic and discuss it with other geeks using [Geeksforgeeks Q&A](#) page. See already discussed [Algorithm questions on forum](#).

Like {318} Tweet {79}  {15}

142 Comments

GeeksforGeeks

 Login ▾

♥ Recommend 11  Share

Sort by Newest ▾



Join the discussion...



Robin Siwach • 2 hours ago

Hey guys I am new to programming. I have noticed that sequence of topics are different in different books. So are the topics interlinked or not i mean can we choose which topic we wanna study or there's sequence we have to follow??

^ | v • Reply • Share >



rohit • 4 days ago



now, it looks better!!

:)

1 ^ | v • Reply • Share ›



Bert • 12 days ago

Hi, I have an array of 1,000,000 short ints. What might be the fastest sort time? I can sort in about 1/50 of a second: 50 million per second. What might be the fastest possible time on a "standard" laptop?

Thanks, Bert

^ | v • Reply • Share ›



Gautham Kumaran • 13 days ago

numbered list available in the blog post :P

[https://itguyninterviews.wordp...](https://itguyninterviews.wordpress.com/)

^ | v • Reply • Share ›



Gautham Kumaran • 18 days ago

@GeeksforGeeks Nice update! numbered list will be so much better

3 ^ | v • Reply • Share ›



sid • 19 days ago

@GeeksforGeeks

how to reset orange color of visited links..?

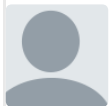
^ | v • Reply • Share ›



DS+Algo → sid • 4 days ago

remove that link from your browser history.

^ | v • Reply • Share ›



Mohammad Nadeem Bhati • 21 days ago

@GeeksforGeeks

in DP Ugly Numbers, Largest Sum Contiguous Subarray are in Same Bullets :)

1 ^ | v • Reply • Share ›



Shubham Gupta → Mohammad Nadeem Bhati • 19 days ago

Thanks for pointing it out. We have corrected it now.

1 ^ | v • Reply • Share ›



Holden • 24 days ago

Thank you for updating! :) It is much better. why some of them are in orange color?

^ | v • Reply • Share ›



Dhruv Gosain → Holden • 24 days ago

the links with orange colour are those which you have already visited.

2 ^ | v • Reply • Share ›



omshanti108 • 24 days ago

FINALLY GEEKSFORGEEKS WEBSITE IS UPDATE :-)

2 ^ | v • Reply • Share ›



Pallavi Singh • a month ago

Hey, can someone please solve this problem for me.

You have 4 types of magic boxes made of ice, of sizes $(1 \times 1 \times 1)$, $(1 \times 1 \times 2)$, $(1 \times 1 \times 3)$, and $(1 \times 1 \times 4)$. Assume that you have an infinite number of boxes of each type.

Using these boxes, you want to make a wall of height N and width M . The wall should not have any holes in it. The wall you build should be one solid structure. A solid structure can be interpreted in one of the following ways: (1)It should not be possible to separate the wall along any vertical line without cutting any magic box used to build the wall. (2)You cannot make a vertical cut from top to bottom without cutting one or more magic box.

The boxes can only be placed horizontally. In how many ways can the wall be built?

Input: The first line contains the number of test cases T . T test cases follow. Each case contains two integers N and M .

Output: Output T lines, one for each test case containing the number of ways to build the wall. As the numbers can be very large, output the result modulo 1000000007.

[see more](#)

^ | v • Reply • Share ›



Dhiraj Soni → Pallavi Singh • 25 days ago

you can see this explanation for a similar problem..<http://stackoverflow.com/questions/16...>
you just have to modify this to get AC :)

^ | v • Reply • Share ›



Raja Taimoor Bagh • a month ago

the values for sorting through bucket sort are... 5,8,4,9,100,1000. how many buckets are required... how we create Intervals?

^ | v • Reply • Share ›



Raja Taimoor Bagh • a month ago

Any body help me about bucket sort?

^ | v • Reply • Share ›



Raja Taimoor Bagh • a month ago

Hello it's me Raja Taimoor Bagh... I'm student of MS(CS). I have problems in my subject of algorithm.

^ | v • Reply • Share ›



Holden • a month ago

what does 'Misc" mean?

^ | v • Reply • Share ›



Gamer Gamer → Holden • a month ago

Miscellaneous

1 ^ | v • Reply • Share ›



Holden → Gamer Gamer • a month ago

Thank you! :)

^ | v • Reply • Share ›



Pallavi Singh • 2 months ago

Hi Guys,

As i don't know about Graph so I started reading the chapter from Algorithm by Steven Skienna, Also tried reading it from Adam Drozdek but I'm not able to understand anything. Is there any other way that I can try?

^ | v • Reply • Share ›



Ajcoo → Pallavi Singh • 11 days ago

watch video lecture series on ds and algo by Dr. Naveen Garg . watch it on youtube or Nptel site.

^ | v • Reply • Share ›



Youssef Yossry → Pallavi Singh • a month ago

First, You need to know some terminologies in the graph theory. then try the most popular sites for Algorithms like:

1- <http://e-maxx.ru/algo/>

2- <https://www.topcoder.com/commu...>

you could also use GeeksForGeeks, it's very useful.

you also should start with the very basic elementary graph algorithms. like:

1- DFS, BFS

2- Topological Sort

- 3- Search for connected components.
 - 4- Minimum spanning tree
 - 5- Dijkstra Algorithm, Bellman-Ford Algorithm.(Single source shortest paths Algorithms)
- and moving towards the complicated
- 1- Flow problems (Edmonds-Carp, Diniz's Algorithms)
 - 2- Connectivity problems
 - 3- Lowest common Ancestor problem
- and so on...

2 ^ | v • Reply • Share ›



Himanshu Dewan → Pallavi Singh • a month ago

U can try Intro to algorithms by Cormen. Its too good.

^ | v • Reply • Share ›



Pallavi Singh → Himanshu Dewan • a month ago

Hey Himanshu, i already have cormen and I tried learning the algo from there but I found it complicated plus they deal with equations, many of which have no explanation for their next step of deduction.

I have seen many amazing reviews for this book which is why I bought it but I'm not too sure if it really serves the purpose looking from the coder's point of view.

^ | v • Reply • Share ›



Abhishek Aadi → Pallavi Singh • 19 days ago

OReilly_AlgorithmsInANutshell or Fundamentals-of-Computer-Algorithms-By-Ellis-Horowitz-1984 these book might help u..

^ | v • Reply • Share ›



Gautam Goyal → Pallavi Singh • 24 days ago

<http://nptel.ac.in/courses/106...>

Please visit the above link and go through graph section in the end. This is probably the best series on data structures available.

^ | v • Reply • Share ›



Ashish Aggarwal → Pallavi Singh • 2 months ago

Try Data Structures and Algorithms Made Easy - Narasimha Karumanchi

^ | v • Reply • Share ›



Lakshmi Narasimha • 2 months ago

<http://www.geeksforgeeks.org/b...>

The above page is not displayed. Can you please help me on this.

^ | v • Reply • Share ›



vaibhav singh • 3 months ago

How to turn off a particular bit in a number?, redirects to <http://geeksquiz.com/algorithm...>

^ | v • Reply • Share ›



sansh • 3 months ago

why it is necessary to return 0 in case int main is our function?what if I use void main

^ | v • Reply • Share ›



Bhuvan → sansh • 3 months ago

As u must be knowing that we need to return a value in case we are using function which has return type other than void.

Now main() is also a function so if its non-void function, u need to return some value.

If you define it as int main() , then return value can be zero or some non-zero value.

Now, if we know the concept of function then we know that if there is a function then there should be some caller for that function so that it can be executed.

Now, the most interesting question is "Who calls main() ???".

Think....

No its not magic that when you run program it starts from main().

Its our Operating System (Windows. Linux , etc) that calls the main() whenever a program is executed

[see more](#)

1 ^ | v • Reply • Share ›



notrohit → sansh • 3 months ago

Then you dont need to return anything.

Void specifies that your funtion doesnt return anything.

In this case, main being your function.

I suggest you read some basics of functions

^ | v • Reply • Share ›



Aditya • 3 months ago

Can i get a program/logic for linear probing in hashing?

^ | v • Reply • Share ›

**sk** → Aditya · 3 months ago<http://en.wikipedia.org/wiki/L...>

^ | v · Reply · Share ›

**hema kadaiah** · 4 months agotime complexity graph for first version
prime test

^ | v · Reply · Share ›

**bhavya** · 4 months ago

how to check worst time complexity for this

n

 $p = (5 * n * n) - 4$ $q = (5 * n * n) + 4$

a=sqrt(p);

b=sqrt(q);

if(a*a==p || b*b==q)

then print

else

not

^ | v · Reply · Share ›

**sai krishna** → bhavya · a month agoconstant $O(1)$ time complexity in any case may it be best worst or average case .
because there are nothing like loops or recursion .

^ | v · Reply · Share ›

**GeeksforGeeks** · 4 months ago

Explore Operating System Concepts and C Programming Language on :

<http://cdynamicprogramming.blog...><http://operatingsystemgeeks.bl...>

^ | v · Reply · Share ›

**Aruna** · 4 months ago

Professor McKenzie is consulting for the president of A.-B. Corporation, which is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

a. Describe an algorithm to make up the guest list. The goal should be to maximize the

sum of the conviviality ratings of the guests. Analyze the running time of your algorithm.

b. How can the professor ensure that the president gets invited to his own party?

^ | v • Reply • Share ›



Aruna • 4 months ago

To within a constant factor, how much time does the following algorithm take, in terms of n ?

```
twoToTheN(n)
```

```
if n == 1
```

```
return 0
```

```
else
```

```
return twoToTheN(n-1) + twoToTheN(n-1)
```

1 ^ | v • Reply • Share ›



AVINASH • 5 months ago

Harry is facing an obstacle in his attempt to save the Philosopher's stone from being stolen by Snape (Harry suspects Snape!). He stands on one corner of a giant wizard chess board with the door to the next obstacle on the diagonally opposite end of the board. Due to magical constraints, Harry can only move one step at a time in any of the four direction. Forward, backward, left & right. Also, this being no ordinary wizard board, if Harry steps on any of the blocks with dark mark (the mark of voldemort), he will instantly die.

Help Harry in his quest by writing a code to find out the least number of steps he requires to reach the door alive. Your code should print the least number of steps required. Make the code as efficient as possible (in terms of time complexity).

Unlike muggle chess, wizard chess boards are magical. They are always square in shape and their sizes can range from a 2X2 board to a 100X100 board. Harry is standing on the block at the top-left corner of the board and the door is on the block at the bottom right corner.

Input:

~~The first line contains a number N. This number is the length of the side of the chess~~

[see more](#)

^ | v • Reply • Share ›



nanotech • 5 months ago

please update here. there are many new post . This is important

5 ^ | v • Reply • Share ›



kaos • 5 months ago

@geeksforgeeks

"Why is Binary Search preferred over Ternary Search?" in searching and sorting section takes to searching sorting quiz... pls correct the hyper link

^ | v • Reply • Share ›



GeeksforGeeks Mod ➔ kaos • 5 months ago

Thanks for pointing this out. We have updated the link.

^ | v • Reply • Share ›



rihansh • 7 months ago

@geeksforgeeks

One can also use selection sort for $O(NK)$ solution and bubble sort too for achieving the same complexity :D .

^ | v • Reply • Share ›



sk ➔ rihansh • 6 months ago

Radix sort tooo..

1 ^ | v • Reply • Share ›



Piyush Kavdia • 7 months ago

Your program is to use the brute-force approach in order to find the Answer to Life, the Universe, and Everything. More precisely... rewrite small numbers from input to output. Stop processing input after reading in the number 42. All numbers at input are integers of one or two digits.

Example

Input:

1
2
88
42
99

Output:

1
2
88

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

[see more](#)

5 ^ | v • Reply • Share ›



Piyush Kavdia ➔ Piyush Kavdia • 7 months ago

thanx piyush.. i got it.. :)

^ | v • Reply • Share ›

[^](#) | [v](#) • [Reply](#) • [Share](#) >**Piyush Baivaw** → Piyush Kavdia • 7 months ago

try reading a few editorials before getting started. the question here is to read the input and print the same until the input value is 42. you cant use the for loop because you dont know the number of inputs that you are going to read. use forever running while loop.(while(1) or while(true)) will work. break from the loop if ur input is 42. otherwise print the input value. you dont need an array here. one variable is enough.

[^](#) | [v](#) • [Reply](#) • [Share](#) >[Load more comments](#)