Getting Started with the Sport of Programming

Wednesday, 23 July 2014

Getting Started with the Sport of Programming

This document is to guide those people who want to get started or have just started with competitive programming.

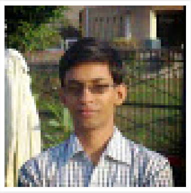
Originally, this document was prepared during the summers of 2014 to help the freshers of Indian Institute of Technology, Kanpur. So, we thought it might be useful to others as well.

Prerequisite : Basics of any programming language. We will follow C/C++.

> **Note : Please note that this blog is not meant to explain concepts in details. The Aim of this blog is to guide you about which topics you should read and practice in a systematic way. However, in many places short explanations have been included for their relevance. Relevant problems are given after each topic. Proper sources are given from where these concepts can be studied. Where sources are not mentioned, that means these are very very popular and you can get to know about them just by a single google search. Move forward and enjoy it !**

All the following things are from our experience and not something written on stone.

About Me



Triveni Mahatha

View my complete profile

Blog Archive

▼ 2014 (1)

▼ July (1)

Getting Started with the Sport of Programming

- You will need to show motivation.
- Languages that should be used
  - C/C++/JAVA (your choice)
  - We will focus on C++, JAVA is slow (one big advantage of JAVA is Big Integers, we will see later)
  - C++ is like superset of C with some additional tools. So, basically if you have knowledge of C, you are ready to code in C++ as well. Otherwise go back and learn how to write codes in C/C++
  - Sometimes knowledge of PYTHON is helpful when you really need big integers.

PARTICIPATE PARTICIPATE PARTICIPATE (the only mantra)

- SPOJ: Its a problem Archive (recommended for all beginners
  - Start with problems having maximum submissions. Solve first few problems (may be 20). Build some confidence. Then start following some good coders (check their initial submissions). Then start solving problems topic wise
  - Never get stuck for too long in the initial period. Google out your doubts and try to sort them out or you can discuss with someone (ONLY IN THE BEGINNING).
  - Before getting into live contests like codeforces or codechef, make sure that you have solved about **50-70** problems on SPOJ.

- CODECHEF: Do all the three contests every month. Do participate in CodeChef LunchTime for sure.
  - Even if you are unable to solve a problem do always look at the editorials and then code it and get it accepted (this is the way you will learn).
  - And even if you are able to do it, do look at the codes of some

good coders. See how they have implemented. Again you will learn.

    - Same point apply to TopCoder and Codeforces as well.

- Codeforces: 4 to 5 short contests of 2 hour in a month (Do them once you develop some confidence).
- TopCoder: Once you have proper experience and you can write codes very fast.

Online Programming Contests:

You write codes and submit them online . The judge runs your code and checks the output of your program for several inputs and gives the result based on your program’s outputs.You must follow exact I/O formats. For example, do not print statements like : “please enter a number”, etc :P

Each Problem has constraints:
Properly analyse the constraints before you start coding.

- **Time Limit** in seconds (gives you an insight of what is the order of solution it expects) -> **order analysis**(discussed later)**.**
- The constraints on input (very imp): Most of the time you can correctly guess the order of the solution by analysing the input constraints and time limit .
- **Memory Limit** (You need not bother unless you are using insanely large amount of memory).

Types of errors you may encounter apart from wrong answer :

- **Run Time Error** (Most Encountered)
    - Segmentation fault (accessing an illegal memory address)
        - You declared array of smaller size than required or you are trying to access negative indices .
    - Declaration of an array of HUGE HUGE(more than 10^8 ints) size -_- .
    - Dividing by Zero / Taking modulo with zero :O .
    - USE gdb (will learn in coming lectures)
- **Compilation Error**
    - You need to learn how to code in C++.

  - USE GNU G++ compiler or IDEONE(be careful to make codes private).
- **Time Limit Exceed (TLE)**
  - You program failed to generate all output within given time limit.
  - Input Files are not randomly generated , they are made such that wrong code does not pass.
  - Always think of worst cases before you start coding .Always try to avoid TLE.
  - Sometimes a little optimizations are required and sometimes you really need a totally new and efficient algorithm (this you will learn with time).
  - So whenever you are in doubt that your code will pass or not .Most of the time it won't pass .
  - Again do proper order analysis of your solution .

Sometimes when you are stuck . Check the running time of other accepted codes to take an insight like what Order of solution other people are writing / what amount of memory they are using.

4 MB ~ array of size 10^6 . Or 2-d array of size 10^3*10^3

Standard Memory limits are of Order of 256MB

Order analysis :

Order of a program is a function dependent on the algorithm you code. We wont go in theoretical details just think Order of program as the total number of steps that program will take to generate output generally a function based on input like O(n^2) O(n) O(log n) .

Suppose you write a program to add N numbers .See the following code.

```
int cur,sum=0;
for(int i=0;i<n;++i)
{
   scanf("%d",&curr);
   sum = sum+curr;
}
```

Total number of computations = n*(1+1+1+1)

n times checking i

n times i++
n times scanf
n times + operating

So total of 4*N.
We remove the constant and call it O(N)
This is the simplest I can explain.You will get further understanding with practice and learning.

You must know running time of these algorithms **(MUST)**
Binary Search -> ?
Merge / Quick sort -> ?
Searching an element in **sorted/unsorted** array -> ?
HCF / LCM / Factorization / Prime CHeck ?

We all know the computation power of a processor is also limited.
Assume 1 sec ~ 10^8 operations per second . (for spoj old server it is 4*10^6).
Keep this in mind while solving any problem.

If your program takes O(n^2) steps and problems has T test cases . Then total order is T*N^2.

For T < 100 and N < 1000 . It will pass .
But for T < 1000 and N < 1000 it wont .
Neither for T < 10 and N < 10000

INT OVERFLOW :
Sum three numbers.
Constraints :
0 < a,b,c < 10^9

```
int main()
{
   int a , b,c;
   scanf("%d %d %d",&a,&b,&c);
   int ans = a + b + c;
   printf("%d",ans);
   return 0;
}
```

This program won't give correct output for all cases as 3*10^9 cannot be stored in INTS you need long long int or unsigned int (4*10^9).
what if 0

Comparing Doubles :

```
int main()
{
        float a ;
        scanf(“%f”,&a);
        if(a == 10 ) printf(“YES”);
        return 0;
}
```

float / double don’t have infinite precision . BEWARE (6/15 digit precision for them respectively)
Try the following problem.
http://www.spoj.com/problems/GAMES/

Standard Template Library (STL):
In your code sometimes you need some Data Structures(**DS**) and some functions which are used quite frequently. They already have lots of standard functions and data structures implemented within itself which we can use directly.

- **Data Structures** (To be discussed in later lectures)
  - Vectors
  - Stack
  - Queue
  - Priority Queue
  - Set
  - Map
- **Functions**
  - Sort
  - Reverse
  - GCD
  - Swap

  - next_permutation
  - binary_search (left + right)
  - max, min
  - pow, powl
  - memset

Now imagine writing codes using these inbuilt functions and data structures . It would be much more simpler now.

What headers/libraries should you include ?

Basically the above functions / DS are in different libraries. So in some cases you may need to include many headers . But you can include everything using just one header.

```
#include <bits/stdc++.h>
```

Try the following problem :
www.codechef.com/problems/ANUUND
Which of the above inbuilt function did you use ?

What if you need to sort an Array of structure ?

You can either make a struct and write compare function for it.(Read more at www.cplusplus.com)Or you can use an vector of pair.

Read This:
http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=sorting

Now you are ready to start competitive programming .
You can continue reading this doc or get started on your own . Good luck :)
First, you must learn the basic and well known algorithms . Not only the algorithm but you must also understand why that works , proof , code it and analyze it . To know what basic algorithms you must know you can read :

- http://www.quora.com/Algorithms/What-is-needed-to-become-good-algorithmist-like-top-rankers-in-Topcoder-Spoj-GCJ
- http://www.quora.com/Algorithms/What-are-the-10-algorithms-one-must-know-in-order-to-solve-most-algorithm-challenges-puzzles
- http://www.quora.com/Computer-Science/What-are-the-10-must-know-algorithms-and-data-structures-for-a-software-engineer

Also read these answers on how to start competitive programming and get good at it.

- http://www.quora.com/ACM-ICPC-1/For-an-ACM-beginner-how-should-I-start
- http://www.quora.com/Can-I-crack-the-ACM-ICPC-in-1-5-years-if-I-have-to-

start-from-scratch

- http://www.quora.com/Competitive-Programming/What-was-Anudeep-Nekkantis-Competitive-Programming-strategy-to-become-35th-in-Global-ranking-in-just-6-7-months

TopCoder has very nice tutorials on some topics, read them here .

You can also read this book topic wise to understand an algorithm in a deeper way http://ldc.usb.ve/~xiomara/ci2525/ALG_3rd.pdf.

To get good at writing fast codes and improving your implementation, you can follow this:
My personal advice is to start practicing on TopCoder . Start with Div2 250 master it then start with Div2 500 master it then move to Div1 250 .Also read the editorials of problem you solve and the codes of fastest submissions to learn how to implement codes in simple and elegant way.Meanwhile keep learning algorithms and keep practicing them on SPOJ or CodeChef or Codeforces . And do read the tutorials, after a time you will realize that the tricks and methods to solve are repeating themselves . We learn from practice only . If you read same thing 5 times in different tutorials then it will not be stored in your short term memory only right .

Below are few topics to start with and problems related to those topic.
They are very basic stuffs and you can learn all you need to know by just googling them out.

"When i will get some time I will try to update and give more details about the topics a newbie should cover."
Try to do all the problems stated below if you are a beginner.

PRIMES

- Prime Check (O(log n) also possible read about miller-rabbin)
- Factorization
- Number of factors
- Sum of factors
- Generating Primes using sieve of eratosthenes
- Bounds on number of primes till N
- Euler's totient function
- **Practice Problems :**
  - http://www.spoj.com/problems/NDIV/

    - http://codeforces.com/problemset/problem/431/B
    - http://www.spoj.com/problems/GAMES/
    - http://www.spoj.com/problems/GCJ101BB/
    - http://www.spoj.com/problems/GCJ1C09A/
    - http://www.spoj.com/problems/MAIN72/
    - http://www.spoj.com/problems/WINDVANE/
    - http://www.spoj.com/problems/NDIV/
    - http://www.spoj.com/problems/PTIME/
    - http://www.spoj.com/problems/NDIVPHI/
    - http://www.spoj.com/problems/NOSQ/
    - http://www.spoj.com/problems/AFS/
    - http://www.codechef.com/MAY13/problems/WITMATH/
    - http://www.spoj.com/problems/CUBEFR/

- Try as many as you can.
- Other things that you can read meanwhile
  - Euler Totient function and Euler's theorem [[READ]]
  - Modulo function and its properties
  - Miller-Rabin Algorithm [[READ]]
  - Extended Euclid's Algorithm [[READ]]
  - Keep exploring STL
  - Prove running time of HCF is O(log n)
  - Try sorting of structures
  - Practice few problems on several Online Judges
  - Try to do + - * operations on large numbers(<1000 digits) using char array (for learning implementation)
  - Number of factors and sum of factors in sqrt(n) time ,Number of primes till N

Basic Number Theory

- Modulo operations and Inverse modulo
- How to compute a ^b % p in O(log b), where p is prime

- Find Nth fibonacci number modulo p [Read Matrix exponential]
- n! % p (what if we have lots of test cases)
- ETF (calculation / calculation using sieve)
- Euler theorem , Fermat's little theorem , Wilson theorem [[READ]]
- nCr % p (inverse modulo) (read about extended euclid algorithm)
- (p-1)! % p for prime p, Use of fermat theorem in Miller-Rabin (Probabilistic) (**miller-rabin.appspot.com**)
- 64 Choose 32 < 10^19 we can precompute till herein a 2 dimentional array [Learn use of the recursive relation : (n+1)Cr = nCr + nC(r-1)]
- Number of ways to traverse in 2D matrix[Catalan Number] (what if some places are blocked ? Hint : DP)
- a^b % c . Given Hcf(a,c) = 1 .And what if Hcf(a,c) ! = 1. [[READ Chineese Remainder Theorem, not used much in competition]]
- Matrix Exponentiation
- solving linear recurrence using matrix exponentiation(like fibonacci)

- Practice problems:
  - http://www.spoj.com/problems/DCEPC11B
  - http://www.codechef.com/MAY13/problems/FTRIP/
  - http://www.spoj.com/problems/FIBOSUM/
  - http://www.spoj.com/problems/POWPOW/
  - http://www.spoj.com/problems/POWPOW2 [[CRT]]

- Go through these tutorials (The listed problems might be tough but do read the tutorial)
  - http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting
  - http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=combinatorics
  - http://community.topcoder.com/tc?

    - module=Static&d1=tutorials&d2=math_for_topcoders
    - http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=primeNumbers

Power of BITS

- Numbers are stored as binary bits in the memory so bits manipulation are alway faster.
- Bitwise 'or' operator : |
- Bitwise 'and' operator : &
- Bitwise 'xor' operator : ^
- Bitwise 'left shift' : <<
- Bitwise 'right shift' : >>
- Memset and its uses using function : sizeof()
- Bitmask and use of Bitmask in Dynamic Programming [[subset DP]]
- Some cool Tricks
    - n = n * 2 :: n = n << 1
    - n = n /2 :: n = n >> 1
    - checking if n is power of 2 (1,2,4,8…) ::checking !(n & (n-1))
    - if x is max power of 2 dividing n, then x = (n & -n)
    - Total number of bits which are set in n = __builtin_popcount(n)
    - setting xth bit of n :: n |= (1<<x)
    - checking if xth bit of n is set :: checking if n&(1<<x) is non zero
- Problem : You are given N numbers and a numbers S. Check if there exist some subset of the given numbers which sums equal to S .What if you are asked to compute the number of such subsets ?
- Practice problems:
    - http://www.spoj.com/problems/SPCO/
    - http://codeforces.com/problemset/problem/114/B
    - More will be added later
- Read this for further knowledge

    - http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=bitManipulation

Binary Search

- Try this : http://codeforces.com/problemset/problem/431/D
- Understand the concept of binary search. Both left_binary_search and right_binary_search. Try to implement it on your own. Look at others implementation.
- sample implementation :

```
int l = 0, r = 10000,  key_val = SOME_VALUE, m;
while (r - l > 1)
{
        m = (l+r) >> 1;
        int val = some_non_decreasing_function(m);
        if(val < key_val) l = m;
        else r = m;
}
if  (some_non_decreasing_function(l) == key_val ) return l;
else return r;
```

// this can be modified in a variety of ways, as required in the problem

- Practice Problems:
    - http://www.spoj.com/problems/AGGRCOW/
    - http://codeforces.com/problemset/problem/431/D [[Learn't something new ?]]
    - http://www.spoj.com/problems/PIE/
    - http://www.spoj.com/problems/TETRA/
    - http://www.spoj.com/problems/KOPC12A/

The Beauty of Standard Template Library of C++

- Vectors in one dimension and two dimension
    - http://www.codechef.com/MAY14/problems/CHEFBM

- solve : http://www.codechef.com/MAY14/problems/COMPILER
  Now use stacks to taste its beauty and solve the following problem too.
  http://codeforces.com/problemset/problem/344/D

- Queue
    - http://www.spoj.com/problems/DONALDO/

- Priority Queue
    - http://codeforces.com/gym/100247/problem/I [[First try without using Priority queue]]

- Set
    - http://www.spoj.com/problems/FACEFRND/ [[First try without using set]]
        - What if I tell you that apart from scanning the input this problem can be done in 2 lines ? Interesting ? Think!

- Map
    - http://www.codechef.com/MARCH13/problems/TOTR/
    - http://codeforces.com/gym/100247/problem/C

Some Practice Problems Before you proceed further

- http://www.spoj.com/problems/DCEPC11B/
- http://www.spoj.com/problems/AGGRCOW/
- http://www.codechef.com/problems/CHEFBM
- http://www.codechef.com/JUNE13/problems/PERMUTE

- http://www.spoj.com/problems/KOPC12A/ (recommended)
- http://www.codechef.com/MAY13/problems/WITMATH/ (recommended)
- http://codeforces.com/problemset/problem/431/D (recommended)
- http://www.spoj.com/problems/SPCO/
- http://www.spoj.com/problems/FIBOSUM/
- http://www.spoj.com/problems/POWPOW/ (recommended)
- http://www.codechef.com/AUG13/problems/CNTSOLS/
- http://www.spoj.com/problems/IOPC_14F/
- http://www.spoj.com/problems/NDIVPHI/ (recommended)
- http://www.spoj.com/problems/AU12/ (easy)
- http://www.spoj.com/problems/ETF/ (easy)
- http://codeforces.com/problemset/problem/114/B (easy)
- http://www.spoj.com/problems/HISTOGRA/ [[Hint : use stacks]]
- http://www.spoj.com/problems/HOMO/
- http://www.spoj.com/problems/NGM2/

GRAPHS

- Try the following problems :
  - Prime Path
  - Prayatna PR

  Any Ideas ?
- Def : Think graphs as a relation between node , related nodes are connected via edge.
- How to store a graph ? (space complexity)

    - Adjacency Matrix (useful in dense graph)
    - Adjacency List (useful in sparse graph) O(min(deg(v),deg(u)))

- You must know the following terminologies regarding Graphs :
    - Neighbours
    - Node
    - Edge
    - Degree of vertices
    - Directed Graph
    - Connected Graph
    - Undirected Graph
    - Connected components
    - Articulation Points
    - Articulation Bridges
    - Tree [[connected graph with N nodes and N-1 edges]]
        - Leaves
        - Children
        - Parent
        - Ancestor
        - Rooted Tree
        - Binary Tree
        - K-ary Tree
    - Cycle in graph
    - Path
    - Walk
    - Directed Acyclic Graph [[DAG]]
        - Topological Sorting (Not very important, in my opinion)

    - Bipartite Graph (Tree is an example of Bipartite Graph . Interesting Isn’t it.)

- Breadth First Search/Traversal (**BFS**) [[very important, master it as soon as possible]]
    - Application : Shortest path in unweighted graphs

- Depth First Search/Traversal (**DFS**) [[very very important, master it as soon as possible]]
    - Infinitely many applications, just kidding :P (But Its true, Indeed !)
- Now try the problems given at the beginning !
- Practice Problems :
    - http://www.codechef.com/JUNE14/problems/DIGJUMP
    - http://www.spoj.com/problems/PRATA/
    - http://www.spoj.com/problems/ONEZERO/
    - http://www.spoj.com/problems/PPATH/
    - http://www.spoj.com/problems/PARADOX/
    - http://www.spoj.com/problems/HERDING/
    - http://www.spoj.com/problems/PT07Z/
    - http://www.spoj.com/problems/NICEBTRE/
    - http://www.spoj.com/problems/CERC07K/
    - http://www.spoj.com/problems/BUGLIFE/
    - http://www.spoj.com/problems/COMCB/
    - http://www.spoj.com/problems/NAKANJ/
    - http://www.codechef.com/IOPC2013/problems/IOPC13N/
    - http://www.codechef.com/IOPC2013/problems/IOPC13G/
    - http://www.codechef.com/IOPC2013/problems/IOPC13C
- **Problem :** You are given a Graph. Find the number of connected components in the Graph.

    - Hint : DFS or BFS.
- **Problem :** You are given a grid with few cells blocked and others open. You are given a cell , call is **source**, and another cell , call it **dest**. You can move from some cell **u** to some another cell **v** if cell **v** is open and it is adjacent to cell **u.** You have to find the shortest path from **source** to **dest**.

    - Hint : Try to think the grid as a Graph and apply some shortest path algorithm. Which one ? You think !
- **Problem :** You are given a Tree. You need to find two vertices **u** and **v** such that distance between them maximum.
    - Hint : Try to do it in O(1) number of DFS or BFS !

GREEDY ALGORITHMS

Greedy Algorithms are one of the most intuitive algorithms. Whenever we see a problem we first try to apply some greedy strategy to get the answer(we humans are greedy, aren't we :P ?).
Read this tutorial for further insight or you can directly attempt the problems most of the greedy approaches are quite simple and easy to understand/formulate.But many times the proving part might be difficult. But you should always try to prove your greedy approach because most the times it happens that you later realise that you solution does not give the optimal answer.

http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=greedyAlg

They are generally used in optimization problems and there exists an optimal substructure to the problem and solutions are generally O(n log n) (sorting) or O(n) (single pass).

Problems List:

- http://www.spoj.com/problems/BAISED/
- http://www.spoj.com/problems/BALIFE/
- http://www.spoj.com/problems/GCJ101BB/

- http://www.codechef.com/problems/FGFS
- http://www.codechef.com/problems/KNPSK
- http://www.codechef.com/problems/LEMUSIC
- http://www.spoj.com/problems/ARRANGE/
- http://www.spoj.com/problems/FASHION/

Q)A thief breaks into a shop and finds there are **N** items weight of **i**th item is **Wi** and cost of **i**th item is **Ci** and thief has a bag of which can carry at most **W** units of weight. Obviously thief wants to have maximum profit . What strategy he should choose if :

Case 1: If he is allowed to take fractional part of items (like assume item to be a bag of rice and you can take whatever fraction of rice you want). [Hint :: greedy])

Case 2:If he cannot break the items in fractional parts. Will now greedy work ? Try to make some test cases for which greedy will fail.

Most of time when greedy fails its the problem can be solved by Dynamic Programming(DP).

DYNAMIC PROGRAMMING [[DP]]

In my view this is one ***the most important*** topic in competitive programming. The problems are simple and easy to code but hard to master. Practice as many DP problems as much possible.

You must go through this topcoder tutorial and you **must** try to solve all the problems listed below in this doc.

(These are basic problems and some with few variations that we feel one **should** know. You must practice other DP problems too)
Problems list:

- http://www.spoj.com/problems/COINS/
- Read about Maximum Sum Subarray [I dint find exact question on any online judge as its very very basic]
- http://www.codechef.com/problems/DELISH

- http://www.codechef.com/problems/KSUBSUM/
- Q)Finding NCR [Using above discussed recursion in math section and DP]
- https://projecteuler.net/problem=18
- Q)Given a matrix filled with numbers.You are initially at upper left corner , you have to reach to the lower right corner.In each step you can either go right or down.When ever you go to a cell you points increase by value of that cell.What is the maximim possible points you can gain?
- http://www.codechef.com/JUNE13/problems/LEMOUSE
- http://www.spoj.com/problems/MAXWOODS/
- http://www.spoj.com/problems/EDIST/
- http://www.spoj.com/problems/ADFRUITS/
- http://www.spoj.com/problems/IOIPALIN/
- http://www.codechef.com/problems/PPTEST/
- http://www.codechef.com/problems/MAXPR
- http://www.codechef.com/problems/LEBALONS
- http://www.codechef.com/problems/DBOY/
- http://www.codechef.com/problems/HAREJUMP

For further advanced topics you can follow topcoder tutorials.
This also might be helpful introduction to competitive programming - Stanford.

If you have any queries / suggestions please contact us.

Abhilash Kumar
abhilak@iitk.ac.in
https://www.facebook.com/abhilash.276

Triveni Mahatha
triveni@iitk.ac.in
https://www.facebook.com/triveni.mahatha

Co ordinators @ Programming club IIT Kanpur [2014-15]
https://www.facebook.com/groups/pclubiitk/

Posted by Triveni Mahatha at 06:51

+30 Recommend this on Google

69 comments:

Jerry 25 July 2014 at 08:42

This comment has been removed by the author.

Reply

Gourav 25 July 2014 at 08:45

well this is going to be very useful guide for beginners like me..please keep updating this post with more necessary practice problems. It would be so helpful to solve exact right niche problems instead of wasting time on non-useful ones. happy coding :)

Reply

Arjun Mayilvaganan 30 July 2014 at 08:55

I truely will follow what you write, whenever you write, on this blog. Keep them coming. Thank you for not being selfish. Happy to see these kind of people.

Reply

RAVI SHANKAR 31 July 2014 at 20:44

good initiative there !!!!

Reply

Gaurav Kalele 17 August 2014 at 09:58

Awesome advise! thank a lot !

Reply

ashwini sawanth 28 August 2014 at 02:44

This comment has been removed by the author.

Reply

VC 7 September 2014 at 02:26

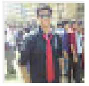
Thanks a ton for this awesome guide! Its difficult to follow the coding groups in college sometimes due to time/projects/other factors, but this guide is a great way to progress in a systematic manner.
What kind of questions should i start with on spoj? Sort by accuracy and solve the ones with highest number of submissions + high accuracy?

Reply

Puneet Singh 4 November 2014 at 18:46

Thanks, its really really concise and informative.

Reply



abhishek shetty 7 November 2014 at 12:16

Hi!,
Given link for topic wise book for algo is not working. Is it the clrs ?

Reply

Anushree 11 November 2014 at 05:05

http://ldc.usb.ve/~xiomara/ci2525/ALG_3rd.pdf is not available. Could you share any alternate link or the book itself?

Reply

Maheep Verma 19 November 2014 at 04:40

This information is very useful to us by the way nice post and thanks for share any way . I want you to visit the link below to get some useful info like Customer Care Number

Reply

Pranav Ganorkar 15 December 2014 at 09:11

Thanks for sharing the info...very helpful...

Reply

Pranav Ganorkar 15 December 2014 at 09:13

Also please keep updating the blog.....

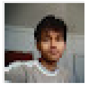
Reply

Andrew Strauss 20 December 2014 at 06:16

Really fantastic discussion. Thank you for sharing this.

SAP HR
SAP Accounting
SAP CRM
SAP Support

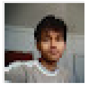
Reply



HEMANT KUMAR 12 January 2015 at 04:00

This comment has been removed by the author.

Reply



HEMANT KUMAR 12 January 2015 at 04:01

Great piece of work....triveni

Reply

Chandrashekhar Kumar 9 April 2015 at 13:52

Great info !

My personal favourite is knuth's volumes along with the book
Cracking Programming Interviews: 500 Questions With Solutions by Sergei Nakariakov

Reply

Chang bui thi 15 May 2015 at 02:37

Thanks for sharing the information. It is very useful for my future. keep sharing
baixar facebook
baixar whatsapp
unblocked games

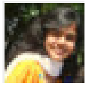
Reply

سواعد لكشف التسربات 31 July 2015 at 09:37

شركة كشف تسربات المياه بالدمام
كشف تسربات المياة بالدمام والاحساء
كشف تسريب المياة بالدمام
كشف تسربات المياة بابقيق
كشف تسربات المياة بابقيق
كشف تسربات المياة بسيهات وعنك
كشف تسربات المياة راس تنورة
كشف تسربات المياة بالدمام والاحساء
كشف تسربات المياة بالدمام والخبر والقطيف والجبيل
كشف تسربات المياة بالقطيف
كشف تسربات المياة بالجبيل

كشف تسربات المياة بالاحساء
افضل شركة كشف تسربات بالخبر

Reply


Shreyasi Verma 4 August 2015 at 12:42

Very useful! thank you so much for sharing this! keep sharing more! (Y)

Reply

Deepak Kumar Gupta 2 September 2015 at 13:12

Thank you sharing your experience. It is very much useful.BTW i was looking for someone who can guide me how to be a good coder.Keep sharing more ideas and thoughts in future...

Reply

Tiffany Lim 3 September 2015 at 23:56

Bavetline
Agen Bola
Agen SBOBET
Agen Judi
Bonus
Prediksi Bola Jitu
Pendaftaran

Reply

دليلك للمعلومات 6 September 2015 at 04:14

social online
كشف تسربات
تحميل كتاب شمس المعارف الكبرى
تحميل لعبة كراش
نموذج سيرة ذاتية

Reply

Darshan Sreenivasamurthy 1 November 2015 at 10:51

Thank you so much :)

Reply

John Adam 2 November 2015 at 07:45

i am very happy to read this article.. thanks for giving us nice info. fantastic walk-through. i appreciate this post.

sports investment

Reply

Hemanth Savasere 3 December 2015 at 07:21

very helpful for beginners who don't know how to start their journey
and well done bro!!

Reply

ankit gupta 8 December 2015 at 02:40

Learn highest paid programming language Earlang tutorial

Reply

philipkotler11 21 December 2015 at 04:37

Thanks for the information and links you shared this is so should be a useful and quite informative!
Body building

Reply

philipkotler11 22 December 2015 at 03:05

Valuable site, where did u come up with the information in this posting? I am pleased I discovered it though, ill be checking back soon to find out what new content pieces u have.
bubble football london

Reply

Jhon Marshal 26 December 2015 at 01:05

I have to say this has been probably the most helpful posts for me. Please keep it up. I cant wait to read whats next.
Body By Vi Results

Reply

Amery Barry 6 January 2016 at 09:30

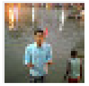
There are certainly a lot of details like that to take into consideration.
zorb football uk

Reply

Rosy James 7 January 2016 at 04:34

Thanks for always being the source that explains things instead of just putting an unjustified answer out there. I loved this post.
the fantasy lineup

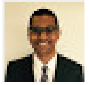
Reply



Prem Kumar 10 January 2016 at 08:56

Sir, I am student of 3rd year btech of electronics and communication engineering from indian school of mines. I want to know that what is real use of competitive programming. I also want to do competitive programming but i have not so much time. Will i become good progrmmer in six months ? Or i leave this and do another thing like php developer or backend developing ?

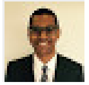
Reply



Dushyanth Kesavan 15 January 2016 at 11:49

This comment has been removed by the author.

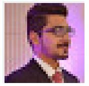
Reply



Dushyanth Kesavan 15 January 2016 at 11:55

The TopCoder tutorials link above doesn't work for me, but this does
TopCoder tutorials

Reply



noman khan 21 January 2016 at 02:47

This point has dependably been one of my most loved subjects to peruse about. I have observed your post to be exceptionally energizing and brimming with great data. I will check your different articles in the blink of an eye. integrated voice response

Reply

hassan emad 21 January 2016 at 02:51

كشف تسرب المياه
شركة كشف تسربات المياه
شركة عزل خزانات

Reply

akmal niazi khan 22 January 2016 at 05:51

This blog awesome and i learn a lot about programming from here.The best thing about this blog is that you doing from beginning to experts level.

Love from

Reply

Edward 31 January 2016 at 08:52

This is too good..
Thanks for sharing this code,.
angularjs training

Reply

Software Programming Tutorial in Roman Hindi/Urdu 9 February 2016 at 12:23

What is diamond problem?

Reply

John Adam 29 February 2016 at 11:16

I like your post & I will always be coming frequently to read more of your post. Thank you very much for your post once more.
edmonton basketball

Reply

Jimmy Kerin 12 March 2016 at 01:11

Resources like the one you mentioned here will be very useful to me! I will post a link to this page on my blog.
zorb football hire

Reply

akmal niazi khan 15 March 2016 at 07:55

Programming is very interesting and creative thing if you do it with love. Your blog code helps a lot to beginners to learn programming from basic to advance level. I really love this blog because I learn a lot from here and this process is still continuing.
Love from Pro Programmer

Reply

Jack Edward 22 March 2016 at 01:51

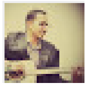
Resources like the one you mentioned here will be very useful to me! I will post a link to this page on my blog.
fu hai feng

Reply

Henry 24 March 2016 at 09:07

One of the best article about programming language,
AngularJs development companies

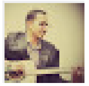
Reply



Anjul Tyagi 28 March 2016 at 21:23

This comment has been removed by the author.

Reply



Anjul Tyagi 28 March 2016 at 21:25

You can see the solutions to above problems in case of difficulty at https://github.com/tyagi-iiitv/Spoj-solutions

Reply

akmal niazi khan 31 March 2016 at 11:55

Programming is very interesting and creative thing if you do it with love. Your blog code helps a lot to beginners to learn programming from basic to advance level. I really love this blog because I learn a lot from here and this process is still continuing.
Love from Pro Programmer

Reply

Jr. Williams 15 April 2016 at 09:08

I think this post will be a fine read for my blog readers too, could you please allow me to post a link to my blog. I am sure my guests will find that very useful.
bubble football

Reply

▼ Replies

Triveni Mahatha ✎ 15 April 2016 at 21:42

Sure. Do post this link if you think people will be benefited.

Reply

Jhon Marshal 18 April 2016 at 00:28

Your Post is very useful, I am truly happy to post my note on this blog . It helped me with ocean of awareness so I really consider you will do much better in the future.
parents

Reply

Jim Marven 28 April 2016 at 10:01

why do west ham sing im forever blowing bubbles?
zorb football hire

Reply

Herry jonson 8 May 2016 at 22:51

wellgooo è l'opzione migliore per l'intrattenimento e la spesa del tempo in questi Attività sport , all'aperto, divertimento , Lago di Gard , attivita , rafting , bici, trekking, escursioni e vela .

Reply

Akatsuki 11 May 2016 at 10:03

thanks a lot

Reply

akmal niazi khan 2 June 2016 at 08:37

Programming is very interesting and creative thing if you do it with love. Your blog code helps a lot to beginners to learn programming from basic to advance level. I really love this blog because I learn a lot from here and this process is still continuing.
Love from Pro Programmer

Reply

David Deans 8 June 2016 at 00:07

Great article it was such an interesting and informative article.
weight training videos

Reply

Toni Smith 15 June 2016 at 22:25

I definitely appreciate your blog. Excellent work and very nice information about the sports.
Skater Owned Skate Shop

Reply

Jhon Marshal 20 June 2016 at 21:49

Hi I really appreciate all the great content you have here. I am glad I cam across it!
Bet

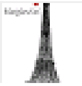
Reply

Jr. Williams 24 June 2016 at 06:11

I like your post & I will always be coming frequently to read more of your post. Thank

you very much for your post once more.
WWE Facts

Reply



محمد الخطيب 3 July 2016 at 21:27

شركة كشف تسربات المياه بالاحساء
كشف تسربات المياه بالاحساء
شركة عزل اسطح بالاحساء
شركة عزل خزانات المياه بالاحساء
شركه عزل مائى بالاحساء
شركة عزل حرارى بالاحساء
شركة ترميم منازل بالاحساء
شركة مكافحة حشرات بالاحساء
شركة رش مبيدات بالاحساء
شركة تسليك مجارى بالاحساء
شركة ترميم منازل بالاحساء

Reply

Manoj Kumar 5 July 2016 at 00:28

nic post...
http://mkniit.blogspot.in

Reply

niaziakmal khan 8 July 2016 at 11:33

Programming is combination of intelligent and creative work. Programmers can do anything with code. The entire Programming tutorials that you mention here on this blog are awesome. Beginners Heap also provides latest tutorials of Programming from beginning to advance level. Be with us to learn programming in new and creative way.

Reply

James Brown 19 July 2016 at 10:48

Great article about the sports and this is such an interesting and informative article.
Quattro - Peg Skate Documentary

Reply

Isabel Bent 19 July 2016 at 13:09

Good information and great post about the sports and i really like it.
Golf Equipment

Reply

YouLoseBellyFat 17 August 2016 at 02:57

we made web site c codes for beginners

Reply

umer 17 September 2016 at 04:38

wow that is so interesting and it's a great information. thanks
Facts about Original ECW

Reply

harrytommy 19 September 2016 at 02:19

Thanks nice comparative description.nice job
Behind The Titantron

Reply

James Brown 28 September 2016 at 02:57

Super blog and very nice and useful information about the sports and wrestling.good work.

Top 10 BEST WWE Wrestling Games

Reply

Mark smith 3 October 2016 at 13:17

Nice article have great information about the play games.
Facts About Wrestling Video Games

Reply

Enter your comment...

Comment as: Select profile...

Publish Preview

Home

Subscribe to: Post Comments (Atom)