



V7

Home | Up

Algorithm Theory

Access 1.5 million documents with
IEEE - Free Trial, Sign Up Today!
www.ieee.org/enterprise

Easy Graph Software

Create Presentation Quality
Graphs See Examples. Try it Free!
www.SmartDraw.com

Say "Hello" to MixMate

The all-around mixer for small
volumes
www.eppendorf.com

Live FTSE100 Trading

Trade CFD's live on the FTSE100
Download our free trading
platform.
www.SaxoBank.com

What Jesus Christ did around 2000 years ago, is simply an act of substitution. We all, like what I said in the past volumes, have gone astray, each of us has sinned. But God, who should punished us because of our sins, has laid on him, Jesus, the iniquity of us all. God substitutes the one who did wrong, us, with Jesus Christ, the one who was recorded as living a sinless life when He was on earth 2000 years ago. Because of this kind of sacrifice by Jesus, we, sinful human, can be declared righteous in front of God since Jesus has paid the cost of our sins... This opens up the access to heaven... The final question is whether we want to take that access or not?... To be continued in [volume 8](#). See previous story in [volume 6](#).

Last updated on: 15 October 2007 08:12:38 PM

Comment on this volume: This volume consists of some regional problems, many of them are still can't be judged. I said that the difficulty rating for this volume is medium...

No	Problem Name	*	Algorithm
700: Mid-Central European Regionals - 1999 (continued to 705-712)			
700	Date Bugs	5.0	Ad Hoc
701	The Archaeologist's Dilemma	6.0	Math
702	The Vindictive Coach	*	Haven't try yet
703	Triple Ties: The Organizer's Nightmare	4.5	Ad Hoc
704	Colour Hash	*	Haven't try yet
705-712: Mid-Central European Regionals - 1999			
705	Maze	*	Haven't try yet
706	LC-Display	5.5	Output-related
707	Robbery	7.0	3-D DFS/BFS..., but I haven't try it yet
708	Dreisam Equations	*	Haven't try yet
709	Formatting Text	*	Haven't try yet
710	The Game	7.0	Runtime Error... something is wrong
711	Dividing up	8.0	DP... but I haven't try yet
712	S-Trees	4.5	Graph
713-721: Central European Regionals - 1998 (2nd link)			
713	Adding Reversed Numbers	3.5	Ad Hoc
714	Copying Books	6.0	Divide & Conquer or DP
715	Substitution Cipher	*	Haven't try yet
716	Commedia dell'arte	*	Haven't try yet
717	Calculating Expressions on Turing Machine	*	Problem description missing...
718	Skycraper Floors	*	Haven't try yet
719	Glass Beads	*	Haven't try yet
720	Hares and Foxes	*	Haven't try yet
721	Invitation Cards	*	Haven't try yet
722-727: East Central Regionals - 1990			
722	Lakes	*	Problem description missing...
723	Comment Removal	*	Problem description missing...
724	Reverse	*	Problem description missing...
725	Division	4.0	Math
726	Decode	7.0	WA... dunno what's wrong
727	Equation	4.0	Math
728-733: East Central Regionals - 1991			
728	Scatter Point Plot	*	Haven't try yet
729	The Hamming Distance Problem	5.0	Ad Hoc
730	Mouse Code Generation	*	Haven't try yet
731	Numerical Summation of a Series	*	Haven't try yet
732	Anagrams by Stack	5.0	Math (Permutation)
733	Follow the Folding Dot	*	Haven't try yet
734-739: Mid Central Regionals - 1993			
734	The Programmer's Hex	*	Haven't try yet

735	Dart-a-Mania	*	Haven't try yet
736	Lost in Space	*	Haven't try yet
737	Gleaming the Cubes	6.5	Ad Hoc
738	A Logical Problem	*	Haven't try yet
739	Soundex Indexing	4.5	Ad Hoc
740: East Central Regionals - 1988			
740	Baudot Data Communication Code	5.0	Ad Hoc
741	Burrows Wheeler Decoder	4.5	Sorting, Decryption
742	Domino Game	9.9	Hm, should be backtracking problem rite?
743	The MTM Machine	4.5	Backtracking
744	Triangular Museum	*	Haven't try yet
745	Numeric Puzzles Again!	*	Haven't try yet
746	Polygon Visibility	*	Haven't try yet
747	Grid Soccer	*	Haven't try yet
748-750: East Central Regionals - 1988			
748	Exponentiation	4.5	Math
749	Machine Repair Simulation	*	Haven't try yet
750	8 Queens Chess Problem	6.5	Chess
751-758: East Central Regionals - 1999 (Minus problem A-B-C-D)			
751	Triangle War	*	Haven't try yet
752	Unscrambling Images	*	Haven't try yet
753	A Plug for UNIX	6.0	Graph (Maximum Bipartite Matching)
754	Treasure Hunt	*	Haven't try yet
755	487-3279	4.5	Ad Hoc
756	Biorhythms	5.5	Ad Hoc
757	Gone Fishing	*	Haven't try yet
758	The Same Game	6.0	Ad Hoc
759-764: Northeast North America Regionals - 1998			
759	The Return of the Roman Empire	7.0	WA, how to handle Roman Numerals??
760	DNA Sequencing	*	Haven't try yet
761	Transform those strings	*	Cannot be judged yet !!!
762	We Ship Cheap	4.5	Graph Traversal
763	Fibinary Numbers	7.0	WA, what's wrong with this?
764	Pentominos	*	Haven't try yet
765-772: Northeastern European Regionals - 1997			
765	References	*	Haven't try yet
766	Sum of powers	*	Haven't try yet
767	Game	*	Haven't try yet
768	Crossword	*	Cannot be judged yet !!!
769	Magic of David Copperfield	*	Haven't try yet
770	Puncher	*	Cannot be judged yet !!!
771	Flying Stars	*	Cannot be judged yet !!!
772	Divide et unita	*	Cannot be judged yet !!!
773	The JustaPox Language	*	Cannot be judged yet !!!
774	Driving in City Squares	*	Cannot be judged yet !!!
775	Hamiltonian Cycle	*	Haven't try yet
776	Monkeys in a Regular Forest	4.0	Graph (Flood Fill)
777	Codebreakers	*	Haven't try yet
778	Recording a tape	*	Cannot be judged yet !!!
779	Wily Hacker's Problem	*	Problem description missing...
780	Sentence Generator	*	Cannot be judged yet !!!
781	Optimisation	*	Cannot be judged yet !!!
782	Contour Painting	6.0	Graph (Flood Fill) + Output-related
783	Trains	*	Haven't try yet
784	Maze Exploration	4.5	Graph (Flood Fill) + Output-related
785-791: Southeastern European Regional - 1996			
785	Grid Coloring	5.5	Graph (Flood Fill) + Output-related
786	Working with Relations	*	Haven't try yet
787	Maximum Sub-sequence Product	9.0	WA, must use BigInteger
788	One Day Tours	*	Cannot be judged yet !!!

789	Indexing	*	Cannot be judged yet !!!
790	Head Judge Headache	*	Haven't try yet
791	Term Reductions	*	Cannot be judged yet !!!
792	Program Modules	*	Cannot be judged yet !!!
793	<u>Network Connections</u>	4.0	Set (Union-Find)
794	Straightest Paths	*	Haven't try yet
795	Sandorf's Cipher	*	Haven't try yet
796	Critical Links	6.5	Graph (Articulation Point)
797	Two Way Traffic	*	Haven't try yet
798	Tile Puzzle	*	Haven't try yet
799	Safari Holiday	*	Haven't try yet

Total submit-able problems in this volume: 100
 Solved problems: 19
 Problems in Wrong Answer list from this volume: 12
 Unattempted problems: 69
 Total hints in this volume: 25

700 - Date Bugs

This problem is 'similar' to 105-The Skyline Problem and 467-Synching Signals. You can use an array of 10000 Boolean flags to mark the years. Try it

703 - Triple Ties: The Organizer's Nightmare

Similar spirit to problem 626, with additional constraint. Just create three nested loops i, j, k from 1 to N . Check for condition:

1. $(i < j < k \text{ or } i > j > k) \text{ and } (\text{win}[i][j] \ \&\& \ \text{win}[j][k] \ \&\& \ \text{win}[k][i])$
2. $(i < j < k) \text{ and } (!\text{win}[i][j] \ \&\& \ !\text{win}[j][i] \ \&\& \ !\text{win}[i][k] \ \&\& \ !\text{win}[k][i] \ \&\& \ !\text{win}[j][k] \ \&\& \ !\text{win}[k][j])$

But since this problem requires you to output the total triples first, you need to do this loop twice. First, to count the total, print it, and then do this $O(n^3)$ loop again to actually print the triples.

Or you can do one loop only, insert all feasible triplets into array, then directly print this array later (faster... but sacrifice more memory storage).

706 - LC-Display

This is a pure output-related problem. Just do what they want, precisely. There are various tricks to solve this problem and it is up to your imagination :)

I can say that solving this problem need patience since you must be very precise...

713 - Adding Reversed Numbers

This problem is easy. Just ignore all 'reverse' stuffs... this problem can be solved without reversing anything...

Read in the input as string!!!, no default data type can store up to 200 digits...

Then do basic carry addition to the right (the normal addition is to align two numbers rightmost and then shift the carry to the left).

Example:

```

4- 3- 5- 8
7- 5- 4-
-----+
11- 8- 9- 8
shift carry to the right
1- 9- 9- 8

3- 0- 5-
7- 9- 4-
-----+
10- 9- 9-
shift carry to the right
0- 0- 0- 1
here... ignore leading zeroes

4- 5-
5- 5-
-----+
9-10-
shift carry to the right
9- 0- 1
here... don't print "9" only (terminate because there is '0' in the middle
but you should print "901"
```

714 - Copying Books

This is a classic partitioning problem. You can either use DP to solve this, or use Divide & Conquer method. More details will be placed here later...

725 - Division

Just systematically enumerate all possibilities... Clever brute force approach will be able to pass the time limit.

727 - Equation

Given an infix expression, convert it to postfix.

There are a lot of Infix to Postfix conversion algorithm available in the web. Go and learn the algorithm to solve this problem. You only need one stack to do this conversion.

729 - The Hamming Distance Problem (by: Wei Tu)

You know the length and how many 1's should be in the bit string. Therefore, you can solve the problem by dual recursion first bit is a 0 or 1, with an extra argument of all the preceding bits.

732 - Anagram by Stack

Permute ii..iioo..oo (total 'i'=total 'o'=length of the original word), and then try simulating this ii..iioo..oo using a stack. If our simulation yields the desired output, then print this permutation.

737 - Gleaming the Cubes (by: Wei Tu)

If there is only one cube, then the total volume is the cube. If there's two, it is the intersection of the two. If there's three, the answer is the intersection of the first one intersects with the third. By keeping the vertices of current intersection cube, you'll be able to solve the problem.

739 - Soundex Indexing

Straightforward conversion will do. Just follow their rules :)

740 - Baudot Data Communication Code

What you have to do is simple, decrypt the input using the given table.

Store the decryption table into an 2 array with size 32. One for Up-Shift table, the other for Down-Shift table. These information are given in the first 2 lines of the input.

Read input per 5-characters, then use your binary->decimal technique to convert them to binary. This is the index for your decryption table.

After that just print out the contents of your array with that index. Print Up-Shift characters if you are in Up-Shift mode, or Down-Shift characters if you are in Down-Shift mode. Use Flag to distinguish these 2 state. Remember: The initial state of each message should be assumed to be in the down-shift state.

741 - Burrows Wheeler Decoder

Solving this problem will be much easier if you understand how Burrows Wheeler compression algorithm works. I suggest that you do Google search on the term: 'Burrows Wheeler'. You'll find the decompression algorithm there. The algorithm is in linear time.

743 - The MTM Machine

You need a recursive checker. Formulate your recursive checker based on the rule given. Once it violates the rule, output "NOT ACCEPTABLE", otherwise, output the new number produced by this MTM machine.

748 - Exponentiation (By: Darkman)

First, found out where the decimal point is. If after the decimal point there are x digits and the power is n, then the final result will have n*x digits after the decimal point (Of course you have to eliminate the trailing zeros explicitly). Then convert the original number into an integer by withdrawing the '.', example, if it 98.876 then the integer is 98876, then use your BigInteger exponentiation. The remaining part is all about printing the output in the right format, which is replacing the '.' back.

750 - 8 Queens Chess Problem

This problem is quite popular, refer to your algorithm books regarding 'backtracking', they usually use 8 Queens problem as a sample.

753 - A Plug for UNIX

A maximum bipartite matching problem. Formulated this problem as a graph and then pass it to a specialized maximum bipartite matching algorithm or a network flow algorithm.

755 - 487-3279

Simulate... and simulate...

756 - Biorhythms

Similar to problem 105,467,and 700... use an array to flags these days... Please browse the internet to find out more about biorhythms.

762 - We Ship Cheap

Solve this problem using Breath-First Search. Formulate the input as a graph, then since the edge weight is uniform, the shortest one found by BFS will be the minimal route. Simply start traversing from starting city to destination city.

776 - Monkeys in a Regular Forest

To solve this problem (+ problem 784 and 785), you need a recursive flood-fill algorithm, which I believe should be a standard algorithm taught in algorithm class.

Flood-fill the area which have the same monkey ID (represent the same family), starting with number 1 from top-left to bottom-right. The output must be formatted as requested, otherwise you'll get Presentation Error.

782 - Contour Painting

Similar as problem 776, we use flood-fill algorithm to paint the maze. But this time we only paint if and only if it is near the border. (the initial '*' can be inside or outside the border, treat them appropriately).

I think this problem is the hardest among 776-782-784-785 flood fill problems...

784 - Maze Exploration

Similar as problem 776 and 782, we use flood-fill algorithm to paint the maze.

785 - Grid Coloring

Similar as 784, just use flood-fill algorithm appropriately. The difference between 784 and 785 is very minimal. You can solve two problems using roughly similar source code.

793 - Network Connections

The best way to solve this problem is using disjoint forest set data structure (implementation of Union Find data structure).

When you know 2 computers are connected, union them by calling `union_set(comp1,comp2)`, then for checking connectivity, you can just determine if `find_set(comp1) == find_set(comp2)`. Everything will be very simple if you do this. However, don't forget if this is a multiple input problems.