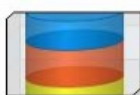




V6

Home | Up



Feedback - Ads by Google

The way God be loving and just at the same time is made possible by the presence of God's son, [Jesus Christ](#), on earth. His presence on earth is historically recorded, it is almost impossible to deny that someone called Jesus has once lived on earth as his life, his incredible miracle acts, his wonderful teachings, his death on the cross, and finally his near impossible resurrection are recorded both in Christian Bible and in many other historical notes. So who is this Jesus Christ? What did He did to allow us to grant access to heaven?... To be continued in [volume 7](#). See previous story in [volume 5](#).

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

Comment on this volume: None

No	Problem Name	*	Algorithm
600-605: Pacific Northwest Regionals - 1997			
600	A Duckpin Tournament	*	Haven't try yet
601	The PATH	5.5	Flood Fill
602	What Day Is It?	5.0	Ad Hoc
603	Parking Lot	*	Haven't try yet
604	The Boggle Game	7.0	WA... can't find the error...
605	The Rotating Disk	*	Haven't try yet
606-613: East Central Regionals - 1998 (2nd link)			
606	Keeps Going and Going and ...	*	Haven't try yet
607	Scheduling Lectures	*	Haven't try yet
608	Counterfeit Dollar	4.0	Ad Hoc
609	Metal Cutting	*	Haven't try yet
610	Street Directions	*	Haven't try yet
611	Parallel Deadlock	*	Haven't try yet
612	DNA Sorting	4.0	Sorting
613	Numbers That Count	*	Haven't try yet
614-619: North Central Regionals - 1997			
614	Mapping the Route	5.5	Ad Hoc
615	Is It A Tree?	6.5	Graph
616	Coconut, Revisited	9.9	WA after rejudge??
617	Nonstop Travel	4.5	Ad Hoc
618	Doing Windows	7.0	Ad Hoc
619	Numerically Speaking	*	Haven't try yet
620	Cellular Structure	4.0	Backtracking
621	Secret Research	2.5	Ad Hoc
622	Grammar Evaluation	4.5	BNF Parser
623	500!	5.5	Math (Big Integer)
624	CD	5.0	Backtracking
625	Compression	*	Haven't try yet
626	Ecosystem	4.0	Ad Hoc
627	The Net	4.5	Graph Traversal
628	Passwords	6.0	Backtracking
629	Test	*	Haven't try yet
630	Anagrams (II)	6.0	Ad Hoc
631	Microzoft Calendar	*	Haven't try yet
632	Compression (II)	*	Haven't try yet
633	A Chess Knight	*	Haven't try yet

634	Polygon	*	Haven't try yet
635	Clock solitaire	*	Haven't try yet
636	Squares (II)	5.0	Math
637-642: Mid-Central Regionals - 1998			
637	Booklet Printing	5.5	Ad Hoc
638	A Duckpin Tournament	*	Haven't try yet
639	Don't Get Rooked	6.0	Chess
640	Self Numbers	5.5	DP
641	Do the Untwist	4.0	Ad Hoc
642	Word Amalgamation	5.0	Anagram
643-648: Pacific Northwest Regionals - 1997			
643	Bulk Mailing	*	Haven't try yet
644	Immediate Decodability	4.5	Ad Hoc
645	File Mapping	*	Haven't try yet
646	The Gourmet Club	*	Cannot be judged yet!!!
647	Chutes and Ladders	5.5	Simulation
648	Stamps (II)	*	Haven't try yet
649	You Who?	*	Cannot be judged yet!!!
650-654: South Central USA Regionals - 1998			
650	Bowl	*	Haven't try yet
651	Deck	*	Haven't try yet
652	Eight	*	Haven't try, Maybe A* will do
653	Gizilch	*	Haven't try yet
654	Ratio	4.0	Ad Hoc
655	Scrabble	*	Haven't try yet
656-664: Southwestern European Regionals - 1998			
656	Optimal Programs	5.5	Ad Hoc
657	The die is cast	5.5	Backtracking, WA after rejudge?
658	It's not a Bug, it's a Feature!	*	Haven't try yet
659	Reflections	*	Haven't try yet
660	Going in circles on Alpha Centauri	*	Haven't try yet
661	Blowing Fuses	3.5	Ad Hoc
662	Fast Food	*	Haven't try yet
663	Sorting Slides	*	Haven't try yet
664	Single-Player Games	*	Haven't try yet
665-672: Northeastern European Regionals - 1998			
665	False Coin	4.0	Ad Hoc
666	Rating	*	Haven't try yet
667	Fence	*	Haven't try yet
668	Parliament	5.0	Ad Hoc
669	Defragment	*	Haven't try yet
670	The dog task	6.0	Graph (Maximum Bipartite Matching)
671	Spell checker	4.0	Ad Hoc
672	Gangsters	*	Haven't try yet
673	Parentheses Balance	3.0	Ad Hoc
674	Coin Change	6.0	DP (CC)
675	Convex Hull of the Polygon	*	No Problem Description...
676	Horse Step Maze	*	Haven't try yet
677	All Walks of length $\leq n$ from the first node	4.5	Graph Traversal
678	Schedule of Taiwan Baseball League	*	Haven't try yet
679	Dropping Balls	4.5	Graph
680	Movement of Reading Head	*	Haven't try yet
681	Convex Hull Finding	7.0	Computational Geometry
682	Whoever-pick-the-last-one-lose	*	Haven't try yet

683	Character Decoding	*	Haven't try yet
684	Integral Determinant	*	Haven't try yet
685	Least Path Cost	*	Haven't try yet
686-693: Asia Regionals (Tokyo) - 1998			
686	Goldbach's Conjecture (II)	4.0	Math (Prime Number)
687	Lattice Practices	*	Haven't try yet
688	Mobile Phone Coverage	9.0	Dunno, Comp. Geometry
689	Napoleon's Grumble	*	Haven't try yet
690	Pipeline Scheduling	*	Haven't try yet
691	Triangle Partition	*	Haven't try yet
692	BUT We Need a Diagram	*	Haven't try yet
693	Digital Racing Circuit	*	Haven't try yet
694-699: North Central North America - 1998			
694	The Collatz Sequence	3.5	Ad Hoc
695	Placing the Ops	*	Haven't try yet
696	How Many Knights	6.0	Chess
697	Jack and Jill	*	Haven't try yet
698	Index	*	Haven't try yet
699	The Falling Leaves	6.0	Graph

Total submit-able problems in this volume: 100
 Solved problems: 33
 Problems in Wrong Answer list from this volume: 14
 Unattempted problems: 53
 Total hints in this volume: 36

601 - The PATH

You can use DFS... but it is a bit troublesome... A better way is to flood fill from left to right, replacing 'w' with some other character, say '1', and check whether in the rightmost column, there exist a '1', if yes, white already have the winning path.

If no '1' exist in the rightmost column, then do flood fill from right to left (opposite direction), replacing all 'w' with another character, say '2'... then check whether there exist an unfilled cell which have '1' on the left and '2' on the right, which means using one more move, white will win...

Same reasoning for black, but this time from top to bottom...

If nothing can be concluded, output there is no winner.

602 - What Day Is It?

Using the Julian & Gregorian date rule, you are asked to determine the day if valid or print error message otherwise. I can say that this is a complex problem and need sufficient knowledge of calendar system. Good luck.

Anyone want to add more explanation for this problem?

612 - DNA Sorting

If you analyze the problem carefully, you'll then realize that "sortedness" here means "how many bubble sort steps we need". After knowing all "bubble sort steps", we can then rearrange the input according to their "sortedness".

Read all input, store it into an array.

Do a bubble sort for each items in array (use temporary variable, sort that variable, not the original, we need them for the output)

Count how many steps we need to make them sorted.

From ``most sorted" to ``least sorted", print them.

614 - Mapping the Route

This problem itself is not difficult, just use the algorithm given in the problem description. The problem is the output format. In my solution, I use a string buffer to store temporary output, modify this output, after everything is ok, I print the string buffer. You must remember that for every movement, you must start from West->North->East->South direction again, not from the last direction.

615 - Is It A Tree?

Check whether the given graph is a tree or not. Refer to your data structures book.

616 - Coconut, Revisited (why WA?)

Just simulate the division process up to sqrt right??... I wonder what's wrong...

617 - Nonstop Travel

Do a brute force complete search from speed 30 mph to 60 mph. To avoid loss precision, use fmod (floating point modulo) instead of mod (integer modulo) and be careful in formatting the output.

620 - Cellular Structure

Do a recursive check of the grammar. Shouldn't be too difficult.

621 - Secret Research

positive result S = 1 or S = 4 or S = 78

negative result S = S35

experiment failed S = 9S4

experiment not completed S = 190S

Actually this problem can be very easy once you know what it means (or when you read this hint). How to know which output we have to give for each input? The answer is simple:

Positive when S is exactly "1", "4", or "78"

Negative when the last two digits of S are exactly "35"

Experiment failed if the first digit of S is exactly "9" and the last one digit of S is exactly "4"

Experiment not completed if the first three digits are exactly "190"

622 - Grammar Evaluation

There is no trick in this problem, just recursively check the expression using the given grammar. Check whether these simple test cases are passed:

```
7
1+2
1++2
1**2
1)(2
(12)
(12)*(12)
(1+2)*(2*3)
```

```
3
ERROR
ERROR
ERROR
12
144
18
```

623 - 500!

A simple factorial problem, and the maximum size is "not 500", but 1000... They said 500 is "too small" :p. If you have your Big Integer library ready, just do it...

624 - CD

Just do an exhaustive backtracking search with effective pruning.

626 - Ecosystem

Max N ?? only 100, simply set a three nested loop i,j,k from 1 to n...

Print the indexes and increase total if $i < j < k$ or $i > j > k$ and i eat j, j eat k and k eat i...

Finally output total possibilities accumulated so far.

627 - The Net

Simple Graph Traversal problem. Either BFS or DFS will do. Just make sure you output the path with lowest IDs if there are more than one minimum path...

630 - Anagrams (II)

If you generate all the anagrams of the word first then, compare it one by one with the dictionary, you will get Time Limit Exceeded.

A better approach is to compare each element in dictionary with the word using my `checkAnagram()` function (see programming section).

636 - Squares (II)

Trial and Error... start from smaller base number one by one until you found the answer.

637 - Booklet Printing

Take two or more pieces of paper and try to make a booklet from these piece of papers manually, then just simulate what human will do to a code.

640 - Self Numbers (by: James Chew)

Dynamic Programming is required here. Prepare a big array of 1,000,000 elements, and then generate all the numbers from bottom up. After that, output the numbers that has no generator (self numbers).

641 - Do the Untwist

Just reverse the formula and decrypt it... straightforward...

642 - Word Amalgamation

This is just an anagram checker problem. Sort the dictionary, record the frequency of each letter in the words. And then when you are given a scrambled word, output all their anagrams, which will be lexicographically sorted by default if you have sort the dictionary.

644 - Immediate Decodability

Brute force... only maximum 8 codes, each code only have at max 10 bits... try all $nC2$ (n Choose 2) possible pairings and check whether bit i and bit j is a prefix of each other... total brute force...

647 - Chutes and Ladders

My childhood game :). This problem is just a simple but tedious game simulation. You must carefully read the problem description (the game rule), and then simulate it.

654 - Ratio

From a given numerator A and denominator B , construct ratios starting with denominator 1 which closest to the actual value of A/B . Keep computing ratios that are getting closer and closer to the actual value until our approximation equal to A/B .

657 - The die is cast

Tricky flood fill problem. After counting the number in a dice, delete that dice to make sure subsequent counting will not count that dice anymore.

663 - Sorting Slides (by: MD Erfan Hoque)

Construct the following graph:
1 to n nodes for the slides, $n+1$ to $n+n$ nodes for the numbers.
edge from the slides to the possible numbers.

Now Run your maximum Bipartite matching algorithm and find all match. For all match edges flip the direction of the edges.
Now start a DFS which start from i node and return back in again at i node [i.e $i=1$ to n]. If for any i it is not possible to return at i then it is correct match to choose the j th number for i th point where i and j is match edge and output this pair.
If no pair found to output then output is none as problem description.

665 - False Coin

A bit similar to problem 608, this problem ask the same question, which one is false/counterfeit coin. If I'm not mistaken you can use the same concept as 608.

670 - The dog task

This problem is a Network Flow (more precise, Maximum Bipartite Matching) problem. Find the best matching for the Bob's positions and interesting places.

You have to construct the bipartite graph first. Add an edge if from Bob's position, the dog can visit the interesting place and go back to Bob without violating the constraint (the dog's round trip distance can't be more 2 times Bob's distance). After that, just pass this graph to your Network Flow or Maximum Bipartite Matching algorithm to get the result.

671 - Spell checker

Very straightforward. Read in the dictionaries (you don't need to sort this...you will have to check all elements anyway...), then read the query words.

If this query word matches any of them, output that this word is correct.

If this query word differs by insert/delete/or change one char, add this to our match list, at the end, output this list.

673 - Parentheses Balance

You have to check whether a given string containing brackets is correct or not. Solution:

1. You can do stack-based solution, pushing every time you encounter open bracket and popping every time you encounter matching close bracket.
2. Or, you can do what I did, Do a looping from 1 to the first occurrence (Forward) of close bracket, then from that point, move backward to the first occurrence of open bracket, then cut that portion of string. Do this until we get empty string (It's a balanced parentheses) or when we cannot do any more search (Unbalanced parentheses).

Example of the second implementation

```
(()) Original string
(()) First search, found ) at index 3
(()) Go back, found ( at index 2
() Delete that portion of string
() Second search, found ) at index 2
() Go back, found ( at index 1
Empty string, it's a balanced parentheses.
```

Please note that empty string is by default correct and there are two types of bracket "()" and "[]", each of them must be treated differently.

674 - Coin Change

Another coin changing problem.

Note:

Coin Changing problem also found in problem 147 (Dollars) and problem 357 (Let Me Count The Ways). You can solve 3 problems using one similar source code.

677 - All Walks of length 2^n from the first node

Simple Depth First Search (bounded to depth 2^n). You don't have to consider the Boolean matrix multiplication given in the problem description, it has no effect. Just don't forget to print "no walk of length n" if there is no walk of length n.

679 - Dropping Balls

A problem that makes you think of efficiency to the limit =). I have to try several times to get this one accepted. Direct simulating the process is impossible since 2^{20} is too big for your memory (and even if you have that memory, you don't pass that annoying 10 seconds Time Limit).

After a careful study of Binary Tree properties... I derive the following formula, which is definitely a cheat if you directly copy this without reading below :p

The formula:

```
P=1;
for (j=0;j<D-1;j++) {
    P=I%2!=0 ? 2*P : 2*P+1; /* go to left or right? */
    I=(I+1)/2; /* round up */
}
```

P is the final value that we want to find (one of the leaves of binary tree), so we loop from 0 to D-1 (yes, this means we compute only the non terminal nodes). For each non terminal nodes, we check whether I is multiples of two, if no, we go to the left (remember, in a tree, left index is $2 \times \text{parent index}$ and right index is $2 \times \text{parent index} + 1$), if yes, we go to the right. Finally we divide the value of I by 2 (rounded up).

Why this formula is true? The root (1st level) will be toggled (yes, TOGGLED!!!, this is the key for efficiency) 1 times, Nodes on the 2nd level will be toggled $1/2$ (rounded up) times, ..., Nodes on kth level will be toggled $1/(2^{(k-1)})$ times. And toggles means binary, and binary is related with modulo 2... so we just check whether I is multiples of 2 or not... =)

681 - Convex Hull Finding

As the name suggests... Your task is "simply" find the convex hull... You can't solve this problem unless you have learned at least one of convex hull finding algorithm (Graham Scan, Jarvis March/Gift Wrapping, Divide & Conquer, Quick hull, etc). Knowing the algorithms is also insufficient since it is very hard (at least for me) to produce a 100% bug-free code... This problem will be a perfect testing ground for those who want to try learning computational geometry :). Try it.

686 - Goldbach's Conjecture (II)

4 until 2^{15} is small :-), prepare a pre-calculated primes below 2^{15} . Then for a given n , just find pairs of primes (from prime table) so that the total of this two primes equal to n .

694 - The Collatz Sequence

$3n+1$ again. It shouldn't be difficult, isn't it? Just simulate it, counting the maximum terms computed.

696 - How Many Knights

There exist a formula to count number of possible placement of knight in $M \times N$ board without backtracking at all. You must derive this formula since 500×500 is too big for simulation.

699 - The Falling Leaves

Given a tree with values in the nodes. Flatten them (i.e. put all nodes on the ground), summing their values. And then print from left to right. The key algorithm to solve this problem is your Tree Traversal algorithm.

This document, vol6.html, has been accessed 12059 times since 27-Dec-00 13:30:12 SGT. This is the 4th time it has been accessed today.

A total of 5951 different hosts have accessed this document in the last 2507 days: your host, 200-71-188-210.genericrev.telcel.net.ve, has accessed it 1 times.

If you're interested, [complete statistics](#) for this document are also available, including breakdowns by top-level domain, host name, and date.