V8

Home | Up

The final question, is whether to believe that Jesus Christ has paid our sins on the cross? While He was on earth, He has told us this statement: "he who believes has everlasting life. I give them eternal life, and they shall never perish; no one can snatch them out of my hand."... He also s aid that he knows God and that He is the way and the truth and the life. No one comes to the Father except through me (Jesus). Previously it was hard for me to accept this story. But knowing that He has actually risen from the dead (and this fact is quite hard to refute), I know that I only have two choices: 1. reject this story, or 2. believe what Jesus said that He is the son of God and He has paid my sins. I have chosen the 2nd option =).

If you have followed my sharing since volume 1 until now and want to know more, you can email me: stevenhalim at gmail.com

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

Comment on this volume: Almost all problems in this volume are from ACM ICPC World Final... Guaranteed to be "very" difficult... and it seems that I haven't solve a lot of problems in this volume...

No	Problem Name	*	Algorithm
800	Crystal Clear	*	Cannot be judged yet!!!
801	Flight Planning	*	Cannot be judged yet!!!
802	Lead or Gold	*	Haven't try yet
803	Page Selection by Keyword Matching	*	Cannot be judged yet!!!
804	Petri Net Simulation	*	Haven't try yet
805	Polygon Intersections	*	Cannot be judged yet!!!
806	Spatial Structures	*	Haven't try yet
807	Towers of Powers	*	Haven't try yet
808-815: ACM World Finals 1999			
808	Bee Breeding	*	Haven't try yet
809	Bullet Hole	*	Haven't try yet
810	A Dicey Problem	*	Haven't try yet
811	The Fortified Forest	*	Haven't try yet
812	Trade on Verweggistan	*	Haven't try yet
813	Robot	*	Haven't try yet
814	The Letter Carrier's Rounds	*	Haven't try yet
815	Flooded!	8.0	Haven't solve this, look at Sohel's notes
816-823: ACM World Finals 2000			
816	Abbott's Revenge	*	Haven't try yet
817	According to Bartjens	*	Haven't try yet
818	Cutting Chains	*	Haven't try yet
819	Gifts Large and Small	*	Haven't try yet
820	Internet Bandwidth	*	Haven't try yet, Network Flow
821	Page Hopping	4.5	Floyd Warshall
822	Queue and A	*	Haven't try yet
823	Stopper Stumper	*	Haven't try yet
...			
824	Coast Tracker	4.5	Ad Hoc
825	Walking on the Safe Side	4.5	DP
826	Symbolic Numerical System	*	Haven't try yet
827	Buddy Memory Allocator	*	Haven't try yet
828	Deciphering Messages	*	Haven't try yet
829	Almost Balanced Trees	*	Haven't try yet
830	Shark	*	Haven't try yet
831	Document Validator	*	Haven't try yet
832	Financial Risk	*	Haven't try yet
833	Water Falls	4.0	Math (Computational Geometry)
834	Continued Fractions	3.0	Math
835	Square of Primes	*	Haven't try yet
836	Largest Submatrix	4.5	DP
837	Light and Transparencies	4.0	Ad hoc

838	Worm World	*	Haven't try yet
839	Not so Mobile	4.5	Backtracking
840	Deadlock Detection	*	Haven't try yet
841	Snake	*	Haven't try yet
842	Crossword Puzzles	*	Haven't try yet
843	Crypt Kicker	9.9	WA
844	Pousse	*	Haven't try yet
845	Gas Station Numbers	*	Haven't try yet
846	Steps	4.5	Ad Hoc
847	A multiplication game	3.5	Math
848	Fmt	*	Haven't try yet
849	Radar Tracking	*	Haven't try yet
850	Crypt Kicker II	9.9	WA
851	Maze	9.9	Must be very efficient, mine TLE
852-860: MIUP 2002			
852	Deciding victory in Go	*	Haven't try yet
853	DVD Subtitles	*	Haven't try yet
854	Worse Code	*	Cannot be judged yet!!!
855	Lunch in Grid City	4.0	Sorting + Median
856	The Vigenere Cipher	*	Haven't try yet
857	Quantiser	*	Haven't try yet
858	Berry Picking	9.9	WA, Computational Geometry
859	Chinese Checkers	*	Haven't try yet
860	Entropy Text Analyzer	*	Haven't try yet
861	Little Bishops	*	Haven't try yet
862	Origami	*	Haven't try yet
863	Process Scheduling	*	Haven't try yet
864	Scheme Pretty-Printing	*	Haven't try yet
865	Substitution Cypher	*	Haven't try yet
866	Intersecting Line Segments	*	Haven't try yet
867	Storing Images in a Sequence	*	Cannot be judged yet!!!
868	Numerical Maze	*	Haven't try yet
869	Airline Comparison	*	Haven't try yet
870	Intersecting Rectangles	*	Haven't try yet
871	Counting Cells in a Blob	*	Haven't try yet
872	Ordering	*	Haven't try yet
873	Loan (II)	*	Haven't try yet
874	2D Representations	*	Haven't try yet
875	Monopoly	*	Cannot be judged yet!!!
876	Balanced Expressions	*	Cannot be judged yet!!!
877	Offset Polygons	*	Cannot be judged yet!!!
878	Rotating Tetris Pieces	*	Haven't try yet
879	Circuit Nets	*	Haven't try yet
880	Cantor Fractions	*	Haven't try yet
881	Points, Polygons and Containers	*	Haven't try yet
882	The Mailbox Manufacturers Problem	*	Haven't try yet
883	Overlapping Rectangles	*	Haven't try yet
884	Factorial Factors	*	Haven't try yet
885	Telephone Directory Alphabetization	*	Haven't try yet
886	Named Extension Dialing	*	Haven't try yet
887	Revolutionary Calendar	*	Haven't try yet
888	Donkey	*	Cannot be judged yet!!!
889	Islands	*	Cannot be judged yet!!!
890	Maze (II)	*	Cannot be judged yet!!!
891	Syntrax	*	Cannot be judged yet!!!
892	Finding words	*	Haven't try yet
893	Y3K Problem	*	Haven't try yet
894	Juggling Trams	*	Cannot be judged yet
895	Word Problem	4.5	Ad Hoc
896	Board Game	*	Haven't try yet

897	Anagrammatic Primes	*	Haven't try yet
898	Hole Cutter	*	Haven't try yet
899	Colour Circles	*	Haven't try yet

Total submit-able problems in this volume: 100
Solved problems: 13
Problems in Wrong Answer list from this volume:4
Unattempted problems: 83
Total hints in this volume: 15

---

815 - Flooded! (by: Sohel Hafiz)

This is a simulation problem. First sort all the heights in ascending order and then greedily fill it with water. First try to pour it in the lowest region and fill it until the level of the water reaches the second lowest region. And then fill the first and second simultaneously until the level reaches that of the third lowest. And then fill 1st, 2nd and 3rd lowest together until the level reaches that of 4th lowest and so on. Stop pouring until you run out of water.

Critical input:
1 1
10
0
0 0

Output:
100% regions is not under water.

820 - Internet Bandwidth (by: MD Erfan Hoque)

One of the easy network flow problem. Build the graph from the given input. Edges are Bi-directional. One trick is that there might be more than one connection between a pair of nodes. At that time add all bandwidth as the bandwidth of that pairs. Then find maximum flow by using any Network flow algorithm. I think there is no critical case if algorithm is ok.

821 - Page Hopping

World final problems... hm... Calculate all pairs shortest path distance (Floyd Warshall), and then output the average. Don't count self edge. Once you can get the sample input correct, most likely you'll get it correct.

824 - Coast Tracker

Starting from the east side (index 6) to north east (index 7) ... 0, 1, 2, 3, 4, until south east side (index 5), check whether that position is a land, if yes, output that direction.

825 - Walking on the Safe Side

This is a DP problem. Let the number of ways at row r, and col c is p[r][c].
when index is (1,1), p[r][c] is 1 (starting point)
when index (r,c) is blocked, p[r][c] is 0
the rest are initialized to -1 (unused).

Then for all unused cells (r,c) (value is -1),
p[r][c] = p[r-1][c] + p[r][c-1];

833 - Water Falls (by: Jagadish)

First, find the uppermost line the drop can fall on.

A drop *d* start from coordinate (x,y). To find the topmost point that this drop d can fall on, we must try all lines. If d's x-coordinate is within a line's leftmost x and rightmost x, then this drop d can (probably) fall on this line. Plug in d's x-coordinate to this line equation to obtain the y-coordinate of the drop. If this y-coordinate is lower than y, then drop d really can fall on this line. Iterate through all lines to pick the topmost line... Then decide, whether to drop will go left or to go right based on line picked.

If such line is not found print the x-coordinate of the drop (on the ground).

834 - Continued Fractions

Store numerator and denominator values, keep simplifying them until numerator becomes 1.

836 - Largest Submatrix

This problem is another variation of 108 (Maximum Sum). If you know how to solve 108, then to solve this problem, you can simply do this:

Convert all 0 to -X where -X is any big negative number, I use -1000

all 1 remains as 1

Then count the rectangle which has the biggest area using the same algorithm for 108 :-)

837 - Light and Transparencies

You don't need Y-axis values at all...
Sort the X-axis coordinates and then use a big array to store all the overall transparency coefficients. Sweep thru all lines, multiple these overall coefficients every time you know a line with transparency coefficient t is above them. Output the result as required.

839 - Not so Mobile

All you need to do is recursively calculate what they want, simple

846 - Steps

Base case:
if x == y, steps = 0

General case:
The most important concept to solve this problem is that the problem description "implies" that the shortest steps must be in a ladder form. 1->2->... increasing -> highest -> .... decreasing -> 2->1. The problem is in determining "highest", since the gap in the middle can be a bit complex. Arithmetic progression formula: n*(n+1)/2 is very helpful here.

To make things easier to understand, I'll use example: x = 1, y = 10

Now, by using Arithmetic Progression (AP) formula from left & right, reduce the gap step by step, until the gap is small enough such that the next AP values will be too big for the gap.

1 2 3 4 5 6 7 8 9 10, difference = 9 (10-1)
1<->2 3 4 5 6 7 8 9<->10, difference = 7 ((10-1) - 2*AP(1))
1<->2<->3<->4 5 6 7<->8<->9<->10, difference = 3 ((10-1) - 2*AP(2))

current AP value = 2

if AP + 1 >= diff, then the difference can be reached by using only 1 next step move
  output 2*AP + 1

but if AP + 1 < diff, then the difference must be reached using 2 steps.
  output 2*AP + 2

847 - A multiplication game

It's quite hard to find this rule... however if Stan and Ollie plays perfect game, then Stan will always try to multiply p with 9 and Ollie will always try to multiply p with 2..., so just simulate the process backwards (i.e. from n, you divide by 9, then divide by 2, by 9... etc until n == 1), then check whose turn can make n becomes 1 and output the winner name.

855 - Lunch in Grid City

Sort streets and avenues, output the median...

892 - Finding Words (by: Saatvik Agarwal)

In this problem all we have to do is remove the punctuation (use ctype.h is alpha()) and take care of the hyphens by putting the hyphenated word on the next line.

895 - Word Problem

For each word in dictionary, count the frequency of each character ['a'..'z']. Then, for each query, also count the frequency of each character. Set total word = 0, then scan through the dictionary one by one, whenever the number of frequency of that particular word in dictionary can be formed using the given query, increase total word by one.

This approach is much faster than enumerating all possible permutation of the query and then check it whether it is inside the dictionary.
