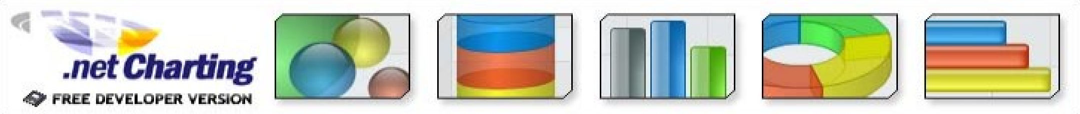
NUS National University of Singapore

V5

Home | Up



God does not like heaven being empty. He said that he have loved us with an everlasting love; that he has drawn us with loving-kindness... However, He cannot simply let sinful man to enter heaven like that. God is holy, clean, pure, sinless. And for this He also said that he does not leave the guilty unpunished as he punishes the children and their children for the sin of the fathers to the third and fourth generation... Wait wait... it seems that there is a conflicting issue here. How can this God be loving and just at the same time?... To be continued in volume 6. See previous story in volume 4.

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

Comment on this volume: Again some regional problems, small amount of world final problems + some local contest problems. Difficulty rating for this volume is medium.

No	Problem Name	*	Algorithm
500-505: Northeastern European Regional - 1996			
500	Table	*	Haven't try yet
501	Black Box	5.0	Ad Hoc
503	Parallelepiped Walk	*	Haven't try yet
504	Random Number	*	Haven't try yet
505	Moscow time	*	Haven't try yet
506-513: ACM ICPC World Finals - 1997			
506	System Dependencies	*	Haven't try yet
507	Jill Rides Again	6.0	DP (Max Interval Sum)
508	Morse Mismatches	*	Cannot be judged yet!!!
509	RAID!	*	Cannot be judged yet!!!
510	Optimal Routing	*	Cannot be judged yet!!!
511	Do You Know the Way to San Jose?	*	Cannot be judged yet!!!
512	Spreadsheet Tracking	*	Haven't try yet, not difficult but tedious
513	Window Frames	*	Cannot be judged yet!!!
514-521: Central European Regionals - 1997 (2nd link)			
514	Rails	3.0	Ad Hoc
515	Kings	*	Haven't try yet
516	Prime Land	3.0	Math (Prime Number)
517	Word	*	Haven't try yet
518	Time	*	Haven't try yet
519	Puzzle (II)	*	Haven't try yet
520	Append	*	Haven't try yet
521	Gossiping	*	Haven't try yet
522-528: Asia Regionals (Shanghai) - 1996			
522	Schedule Problem	*	Cannot be judged yet!!!
523	Minimum Transport Cost	9.9	TLE !!!... how to optimize?
524	Prime Ring Problem	4.5	Backtracking
525	Milk Bottle Data	*	Cannot be judged yet!!!
526	String Distance and Transform Process	7.0	DP (Edit Distance)
527	The partition of a cake	*	Haven't try yet
528	The Problem of Train Setout	*	Cannot be judged yet!!!
529-536: University of ULM Local Contest - 1997			
529	Addition Chains	5.5	Backtracking
530	Binomial Showdown	6.5	Math
531	Compromise	5.0	DP (LCS)
532	Dungeon Master	4.5	Graph Traversal
533	Equation Solver	5.5	BNF Parser
534	Frogger	6.0	Floyd Warshall (Minimax)
535	Globetrotter	5.5	Math (Computational Geometry)
536	Tree Recovery	5.0	Graph

537-544: University of ULM Local Contest - 1998			
537	Artificial Intelligence?	4.0	Ad Hoc
538	Balancing Bank Accounts	4.0	Ad Hoc
539	The Settlers of Catan	4.0	Backtracking
540	Team Queue	5.5	Ad Hoc
541	Error Correction	1.5	Ad Hoc
542	France 98	5.5	Ad Hoc
543	Goldbach's Conjecture	4.0	Math (Prime Number)
544	Heavy Cargo	6.0	Floyd Warshall (Maximin)
545	Heads	9.9	WA, isn't this should be similar as 474 ?
546	Image Recognizer	*	Cannot be judged yet!!!
547	DDF	5.0	Ad Hoc
548	Tree	*	Haven't try yet
549	Evaluating an Equations Board	*	Haven't try yet
550	Multiplying by Rotation	5.5	Math
551	Nesting a Bunch of Brackets	4.5	Ad Hoc
552	Filling the Gaps	*	Haven't try yet
553	Simply proportion	*	Haven't try yet
554	Caesar Cypher	4.5	Ad Hoc
555	Bridge Hands	4.5	Card
556	Amazing	3.5	Simulation
557-564: Northwestern European Regionals - 1996			
557	Burger	5.0	Math
558	Wormholes	6.5	Graph (Bellman Ford)
559	Squares (II)	*	Haven't try yet
560	Magic	*	Haven't try yet
561	Jackpot	*	Haven't try yet
562	Dividing coins	5.0	Math
563	Crimewave	*	Haven't try yet
564	Gaston	*	Haven't try yet
565-571: South Central USA Regionals - 1997			
565	Pizza Anyone?	*	Haven't try yet
566	Adam's Genes	*	Haven't try yet
567	Risk	4.0	Floyd Warshall
568	Just The Facts	5.0	Math
569	Horse Shoe Scoring	*	Haven't try yet
570	Stats	*	Haven't try yet
571	Jugs	4.5	Backtracking
572-577: Mid-Central USA Regionals - 1997			
572	Oil Deposits	3.5	Graph (Flood Fill)
573	The Snail	3.5	Ad Hoc
574	Sum It Up	4.5	Backtracking
575	Skew Binary	3.5	Ad Hoc
576	Haiku Review	4.0	Ad Hoc
577	WIMP	*	Haven't try yet, so complicated
578-584: East Central Regionals - 1997			
578	Polygon Puzzler	*	Cannot be judged yet!!!
579	Clock Hands	2.5	Ad Hoc
580	Critical Mass	4.0	Math (Number Theory)
581	Word Search Wonder	7.0	Complex graph construction + traversal
582	Randomly Wired Neural Nets	*	Cannot be judged yet!!!
583	Prime Factors	3.5	Math (Prime Number)
584	Bowling	4.5	Simulation
585-593: Western and Southwestern European Regionals - 1997 (2nd link)			
585	Triangles	7.0	DP
586	Instant Complexity	4.5	Ad Hoc
587	There's treasure everywhere!	3.0	Ad Hoc
588	Video Surveillance	*	Haven't try yet, a formula exist...
589	Pushing Boxes	*	Haven't try yet, 2 BFS traversal...
590	Always on the run	6.0	Graph Traversal

591	Box of Bricks	1.5	Ad Hoc
592	Island of Logic	*	Haven't try yet,anyone want to explain?
593	Mbone	*	Haven't try yet.. Network simulation??
594-599: Greater New York Regionals - 1997 (Minus problem G)			
594	One Little, Two Little, Three Little Endians	3.5	Ad Hoc
595	A Major Problem	*	Haven't try yet, I'm very weak in Music
596	The Incredible Hull	*	Cannot be judged yet!!!
597	Last Name First, Please	*	Cannot be judged yet!!!
598	Bundling Newspapers	4.0	Backtracking
599	The Forrest for the Trees	*	Haven't try yet

Total submit-able problems in this volume: 100
Solved problems: 40
Problems in Wrong Answer list from this volume:11
Unattempted problems: 49
Total hints in this volume: 45

501 - Black Box (by: Alexander Dolin)

Use two heap data structures, one is maximum heap (heap1) and the other is minimum heap (heap2). At the step *i*, we have to find the number with order statistic *i* in the final sorted array. So, we keep first *i-1* numbers in heap1, other numbers in heap2. Minimum from heap2 will be the answer.

507 - Jill Rides Again

The underlying algorithm for this problem is a "maximum interval sum", and there is a nice linear time DP algorithm to solve this problem (yes only linear time algo can pass the time limit, since the problem size can be as big as 20000 'stops'.

```
Niceness[i] = Niceness[start] + ... + Niceness[i]
                if sum from index 'start' to i is >= 0 or
              Niceness[i], set start=i+1 (start new interval)
                if sum from index 'start' to 'i' < 0
```

The simple reasoning of this DP formulation is as follows: if you have positive (or zero) sum, then this current sequence can still be extended to a longer interval with bigger value or at least similar value but longer interval... but if the partial sum is negative... then there is no point to extend it further...

Example from sample input:

```
Niceness: -1 6
Sum     : -1 6
             ^
          max sum

Niceness: 4 -5  4 -3  4  4 -4  4 -5
Sum     : 4 -1  4  1  5  9  5  9  4
             ^                 ^
            stop            max sum

Niceness: -2 -3 -4
Sum     : -2 -3 -4
           ^
        max sum, but negative... no nice parts
```

So, just do a linear sweep from left to right, accumulate the sum one element by one element, start new interval whenever you encounter partial sum < 0... At the end, output the longest and most nicest, "j-i" interval.

514 - Rails

Using only one-end station (Hint: a Stack), you must determine whether it is possible to marshal the coaches in the order required on the corresponding line of the input file.

Output "Yes" if it is possible, otherwise output "No". Solution:
1. Use a stack.
2. Trial & Error using a piece of blank paper first, then you'll see the pattern.

Common Mistake:

1. Input can be like this

> 5
> 1 4 3 2 5
> 0
> 0
>
> And the output for this is "Yes".

2. Incorrect stack implementation, an 1000-elements array is sufficient.

516 - Prime Land

You are given a "Prime representation" of an integer number X | 2<X<=32767.
Then you have to decrement X by 1, and then output the value of X in its new "Prime representation"

Example of "Prime representation":

Let X=5, then Prime representation of X is 5^1 (Written "5 1")
Let X=10, then Prime representation of X is 5^1 * 2^1 (Written "5 1 2 1")
Let X=100, then Prime representation of X is 5^2 * 2^2 (Written "5 2 2 2")

So "Prime representation" is the form of product of powers of prime factors. There will only be one way to represent X in its "Prime representation" for all X>1. Solution:

1. Convert "prime representation" to an integer X, multiply the powers of prime factors of X.
2. Decrement X by 1.
3. Turn X into its "Prime representation" again. Use prime factors algorithm
(See my programming page) ~> (Similar to number 583).

524 - Prime Ring Problem (with help from: Arief, Lucas)

This problem can be solved using efficient backtracking. Even though n is "just" 16, finding the combination of "prime ring" can be as big as 16! if you do brute force. Prune whenever you can.

526 - String Distance and Transform Process

Although Edit Distance DP algorithm is quite popular. It's a bit hard to tweak the code to get it accepted by the judge. I can only say good luck in tweaking your Edit Distance / Approximate String Matching algorithm.

529 - Addition Chains

Again, this is another backtracking problem. Always remember the rule of thumb: "Prune whenever you can"

530 - Binomial Showdown (by: Felix Halim)

This is just standard nCr (Combination) calculation, where nCr = n! / (r!*(n-r)!).
But this one uses very large numbers and you are likely to get overflow, or time limit.

It's up to you to design any algorithm that can solve this.
However, the basic idea is how to make algorithm like this:

1. Simplifies n! / (r!*(n-r)!) to simpler form.
   Example: 5C2 = 5! / (2!*(5-2)!) = 5! / (2!*3!) = 5*4*3*2*1/2*1*3*2*1 = 5*2
2. And then multiply the simplest form of nCr (5*2) = 10
3. Output the result. Using this trick, you will not get overflow error.

531 - Compromise

If the normal LCS compare characters, this version compare strings..., just re-use your LCS algorithm and adjust it to compare strings... done

532 - Dungeon Master (by: Felix Halim)

2-D maze problems are very familiar. This problem is similar, but in 3-D. Fortunately, you don't need to worry much about the complexity of moving to 3-D space..., you can simply re-use your BFS code without major modifications.

533 - Equation Solver

A bit complex... Given the grammar of the math in BNF, calculate the unknown variable. You can simulate this using elementary school technique..., quite troublesome, I know, but doable...

534 - Frogger

A frog's jump range is must be at least as long as the longest jump occurring in the sequence.

The frog distance (minimax distance) between two stones therefore is defined as the minimum necessary jump range (NOT total jumps) over all possible paths between the two stones.

Example:

2
0 0
3 4

Output -> 5.000, direct jump from stone freddy stone to fiona stone

```
3
0 0
3 4
3 0
```

Output -> 4.000

Jump from freddy stone (0,0) to intermediate stone (3,0), range->3.000, then jump from intermediate stone (3,0) to fiona stone (4,0), range->4.000. The longest jump in this sequence is 4.000, therefore the jump range for this sequence is 4.000.

This sequence is smaller than direct jump (example 1 above) which is 5.000, so, for this test case, you output 4.000 (minimum necessary jump range).

Algorithm: All Pairs Shortest Path, for example: Floyd Warshall algorithm.

535 - Globetrotter

Use spherical / geometrical distance formula. Read morehere.

536 - Tree Recovery

You are given two representation of a tree, the preOrder and inOrder representation.
You have to re-build the tree and output the tree in postOrder representation.

Use Tree Recursion, recursively partition the string based on this fact:
the first element of preOrder is the root, find this root in inOrder representation, partition the string according to that root, recursively.

537 - Artificial Intelligence?

This problem is basically simple, compute P=U*I or U=P/I or I=P/U. However, parsing the input can be harder than the problem itself :). Master your programming language I/O skill in order to parse the input correctly...

538 - Balancing Bank Accounts

Sort the input and greedily assign the money properly...

539 - The Settlers of Catan

Simple backtracking will solve this problem. Just explore everything... The number of node and edges are small (less than 25).

540 - Team Queue

Even though the problem description is clear and should be easy... The size of input will be the real problem. "In constant time" may be impossible (not sure)... but binary search (log n) is sufficient (I get accepted). This problem can be a good test for testing how efficient your code is (in terms of memory and speed).

541 - Error Correction

Error correction mainly used in Computer system's memory management. There are even parity and odd parity. In this problem, we have to check even parity.

Count the number of ``1'' for each rows and columns, all of them must be even.
If there exist one or more error, do this:

If the error is on the same ROW and COLUMN, then output "change bit (row,col)"
else output "corrupt"

543 - Golbach's Conjecture

Simulate this: "Every even number greater than 4 can be written as the sum of two odd prime numbers."

Store a prime list in array (up to n), find the pair (If there is more than one pair of odd primes adding up to *n*, choose the pair where the difference *b* - *a* is maximized.)

This line is, however, will not be executed...(If I'm not mistaken) (If there is no such pair, print a line saying ``Goldbach's conjecture is wrong.'')

544 - Heavy Cargo

Whenever you encounter a phrase like "maximize the minimum" in a problem statement... you can guess that the problem has to do with All Pairs Shortest Path, Floyd Warshall maximin variant. Try it.

551 - Nesting a Bunch of Brackets

Use a stack, push when you encounter open bracket, pop when you encounter close bracket. Errors will occurs if the popped item is not matching with current close bracket, or when at the end, the stack is not empty...

555 - Bridge Hands

Simple card simulation. Simply deal those card to 4 piles (remember starting position, it can be from North, East, South or West). After that, sort the piles according to this problem rules.

556 - Amazing

A simulation of robot movement..., just do according to problem description.

557 - Burger

Refer to your discrete mathematic books (probability theory)

Probability a child get a hamburger => (1/2)^x,
where x=people-2 because we want to keep Ben & Bill get the same burger
since this flipping is done sequentially, first child get (1/2)^x,
second child get (1/2)^x * (1/2)^x, and so on...

558 - Wormholes

Construct the graph, and then pass this graph to your Bellman Ford Shortest Path algorithm. Bellman Ford can detect the presence of negative cycle, and this is what you want to know...

562 - Dividing Coins (by: Abdullah Al Mamun)

Use one dimensional, left to right traversal, dynamic programming.

567 - Risk

This is an All Pairs Shortest Path problem. However, since total vertex is small (at most 20 countries), a simple brute force DFS/BFS will do...

571 - Jugs

Another backtracking problem. You have 6 branching factors (6 type of moves). Perform this backtracking by disallowing repeated cycles (by storing a flag in memory that you already visit a similar jugs configuration before). The main problem here is just the Time Limit...

572 - Oil Deposits

Another backtracking problem... (There are a lot of backtracking problem in this volume). Starting from a particular '@' cell, flood fill it to 8 directions..., then find the number of components.

573 - The Snail

A simple problem, but there are several traps:

1. Beware when fatique<0, the snail will not fall down again
2. If the snail already manage to get out, don't come back !!!

574 - Sum It Up

Again..... another backtracking problem. Backtrack, prune, backtrack, prune...

575 - Skew Binary

Base number, but 'skewed'... so, use the new rule to convert the binary -> decimal.

576 - Haiku Review

Hm... just follow the rules... I can't tell much...

579 - Clock Hands

You have to determine the angle between 2 clock hands.

Use this simple algorithm:

```
hAngle=h*30+(m/60)*30; // Angle from 12o'clock to hour hand
mAngle=m*6;            // Angle from 12o'clock to minute hand
angle=abs(hAngle-mAngle);
if (angle>180) angle=360-angle;
```

Common Mistake

1. Forget to subtract the angle with 180 if it is larger 180 (They want the smallest angle)
2. This is 12 hour clock !!! Clock with hands usually 12-hour clock !!!

580 - Critical Mass (by: Rupam)

n L's or U's can stacked up 2^n ways. Say, there is x ways of arranging stacks in which, there are no more than 2 consecutive U's. Then, number of ways stacks can be arranged in which there is at least one occurrence of three consecutive U's, can be written as:
$C(n) = 2^n - x$.

Now, x can be denoted as $A(n)$, number of ways of arranging stacks of n L's or U's in which, there are no more than 2 consecutive U's and lets name this type of arrangement: *B type*.

Then n things of *B type* can be seen as:

```
_______________ = ________________L + _______________LU + ________________LUU
n things of B type   n-1 things of B type   n-2 things of B type   n-3 things of B type
```

The right hand side permutations makes n things of B type as well, so both sides are equal.
Therefore, $A(n) = A(n-1) + A(n-2) + A(n-3)$
where, $A(1) = 2, A(2) = 4, A(3) = 7$
these are base cases for *B type* when n = 1,2, and 3

In mathematic, there is a special series called "Tribonacci series", where
$T(n) = T(n-1) + T(n-2) + T(n-3)$
with base cases:
$T(1) = 1, T(2) = 1, T(3) = 2$

Our series A(n) can be transformed to Tribonacci series => $A(n) = T(n+2)$
Therefore:

$C(n) = 2^n - x$
$C(n) = 2^n - A(n)$
$C(n) = 2^n - T(n+2)$

So, just implement this $C(n)$ formula :)

Actually, whoever understands the problem this way, before knowing that the solution is $2^n - T(n+2)$ can go easily without knowing Tribonacci series, using A(n).

583 - Prime Factors

This problem wants us to convert a number to its Prime factors. Very similar to 516, but this one is simpler. Solution:

1. Take the input X.
2. Start with a counter=2.
3. If counter can properly divide X then print counter and divide X by counter.
4. If counter>=sqrt(X), stop and print X directly, remember divisibility property.
5. If X=1 then stop else go back to step 3.

Common Mistake

1. Remember what to print for negative numbers and when to print " x " symbol.
2. Time limit. Remember this divisibility property: No number > sqrt(X) can properly divide X.

584 - Bowling

Another simulation problem. There are many ways to solve this problem, pick the one that is easiest for you. Familiarize yourself with bowling scoring rule as described in problem description. There is no trap in this problem. As long as you can model the scoring rule in your code, you'll get accepted.

587 - There's treasure everywhere!

Simply move according to the input data. At the final destination, just compute the distance between final destination and point (0,0) using standard phytagoras calculation.

Start with initial value x=10e-12 && y=10e-12, I don't know why, but using this trick your program will get accepted, otherwise, you will possible got WA.

589 - Pushing Boxes (by: Arif Uzzaman)

Two bfs functions are needed. One for you and one for box. But you have to be careful on some point.

In bfs function for the box:
- in general bfs algorithm one node is visited only once but in this case a node can visited more than once.
- the box can visit a node from east once, from west once, from south once and from north once. so a node can be visited by a box 4 times except the initial node for the box, it can be visited at most five times.

Some critical inputs:

```
12 11
###########
#.#.......#
#.###..####
#...#...#.#
#####.##..#
#.....#...#
##.##...###
#....B..T##
#...#.#####
#####.S..##
##...######
###########
12 11
###########
#.#.......#
#.###..####
#...#...#.#
#####.##..#
#...#.#...#
##.##...###
#....B..T##
#...#.#####
#####.S..##
##...######
###########
12 11
##.....####
#.........#
#.###.#.###
#...#...#.#
#####.##..#
#...#.#...#
##.##...###
#....B..T##
#...#.#####
#####.S..##
##...######
###########
```

output:

```
Maze #1
wnNwwwnneeeSnwwwsseeEEE

Maze #2
wnNNNNNennwSSSSSesWWWswwnEEEEEE

Maze #3
wnNNNNNeennwwSSSSSesWWWswwnEEEEEE
```

591 - Box of Bricks

There are a lot of programming problem similar to this one, memorize this useful technique (If you want).

Example: 5 2 4 1 7 5

Sum all items => 5+2+4+1+7+5=24
Find the average value => 24/6=4
Do a looping from first item, count the differences from the average, get the absolute value
5-4 = 1 => 1
2-4 = -2 => 2
4-4 = 0 => 0
1-4 = -3 => 3
7-4 = 3 => 3
5-4 = 1 => 1
Sum the absolute difference => 1+2+0+3+3+1=10
Divide by 2 (because you don't have to do it twice, think about it) => 10/2=5
Output the result = 5

594 - One Little, Two Little, Three Little Endians

You need to swap bits!

The input is an integer N, convert this to 32-bit integer. You have to swap 8 bits of Least Significant Bit to Most Significant Bit. Partition them into this:

X1 X2 X3 X4

Where X1,2,3, & 4 is 8-bit from the complete 32-bit integer.
Then you need to swap it so the position is like this:

X4 X3 X2 X1

So, use bitwise manipulation

<< Shift Left
>> Shift Right
& bitwise And

The implementation is up to you.

598 - Bundling Newspaper

This is a simple backtracking enumeration problem. The only problem that you may encounter is in reading the multiple input format precisely. Other than that, this is just a simple brute-force enumeration problem using backtrack. Btw, no need to sort the newspaper names. The terms "lexicographic" in the problem refer to the notation A,B,C,D given in the problem, that is, you enumerate using the order given in the input, not based on the newspaper name.
