# Discussion On My solved ACM Problem

# 10583

# This problem may be solved by union find data structure. But yet I have

# Not studied about this data structure ,so I tried it other way. But after analyzing the problem

# Carefully ,I understood this problem actually wants the number of disconnected components. So I applied BFS repeatedly every unexplored node .DFS will do also.

# 10926

# I simply solved it applying DFS every unexplored node and found out maximum number from which node other node can be reached.

# 11222

# Declare a two dimensional array,mark 1 the solved problem by each person.Then count linearly for each person which columns is unique..and count the total unique column.

# Be vary careful about the size of array limit.I got runtime error for 4 times in this problem.Because I declared array with 10001 then 20000 etc.Finally accepted limit was arr[10][50000].

# 441

# Consider a case: 8 1 2 3 4 5 6 7 8

# Problem asks to print all possible combination of 6 numbers from above case.Now think for first number 1,how many possible combination may exist…Its obviously (8-5)=3 combination.Its also true for second number,3rd,4th,5th,and 6th number etc.So just run six nested loop each one running for <k-5 times.Try it using completely with brute force method and if you fail then look at below snippet.

# for(i=0;i<n-5;i++)

# for(j=i+1;j<n-4;j++)

# for(k=j+1;k<n-3;k++)

# for(l=k+1;l<n-2;l++)

# for(m=l+1;m<n-1;m++)

# for(o=m+1;o<n;o++)

# 567

# All pair shortest path problem.Use modified floyed warshall.Be careful about Output manner..The program wants a blank line after results printed out after each test case..it does not want any blank line between consecutive inputs.

# 530

# Sample example how I solved:

# For n=10 r=6 and n-r=4,then I declared an array to store values greater than max of n or (n-r) upto n.

# So for this case a[0]=7,a[1]=8,a[2]=9,a[3]=10

# Then a[0]=7,a[1]=8/4=2,a[2]=9/3=3,a[3]=10/2=5

# Then multiply this numbers and output.If you are getting wrong answer or TLE

# Then try this Sample input:

# 1234567 0

# 4567835 4567835

# 1234567890 1

# 1000000000 999999999

# 0 0

# Sample Output:

# 1

# 1

# 1234567890 //Should not take much time

# 1000000000

# 644

# I noticed a thing here while coding for this in c:

# While(scanf(“%s”,a[0])==1)

# {

# While(1)

# {

# }

# Printf(“\n”);

# }

# Above code snippet print a blank line for EOF…but it is not a problem in c++.I don’t know why.

# 10126

# 1.Use quick sort to sort the words in alphebatical order.

# 2.Tokenize word very carefully.Note “it,is me”-here three words like ‘it’,’is’,’me’

# 3.Dont use function strcmpi().Its not supported in ACM.

# 4.qsort((void\*)list,n,sizeof(list[0]),compare\_fun);

# 10298

# I solved it using Brute force method.Clever checking may pass TLE.

# 423

# The problem wants to compute minimum time needed to pass a message to all other processor from first processor.So you may apply Floyed Warshall’s algorithm to compute shortest path from first processor to all other processor.But tricks is here that problem wants time needed to sent mesagge all the processor.So find out maximum of all shortest time from first node;

# 280

# Simply apply DFS and print which nodes are inacsessible from given starting node.Remeber there may be self-edge.Think which represtation is better for this problem.

# 336

# Hmmmm…tedious program…I got CE 4 times because I didn’t noticed that

# #define SIZE 50000

# For(i=0;i<=SIZE;i++)

# Here array is overruning.So CE.Then 2 times WA.well..i’ll not expalin why I got WA.just guess it from below test cases…and u can solve it using BFS.

# 2

# 1 2 4 5

# 1 1

# Case 1: 2 nodes not reachable from node 1 with TTL = 1.

# 3 2

# Case 2: 4 nodes not reachable from node 3 with TTL = 2.

# 2 0

# Case 3: 3 nodes not reachable from node 2 with TTL = 0.

# 459

# The problem wants number of components in the graph.I used DFS to solve this problem.Though problem wants input to be terminated by reading a blank line..I got TLE trying it…so use EOF to end input taking..and consider following cases:

# Sample Input:

# 3

# A

# C

# BA

# Z

# Sample Output:

# 1

# 2

# 26

# 11690

# Read the problem carefully.Add money of the people who have realationship and check whether total amount is zero ..then set 0 to people of that relationship.If everyone’s money set to 0 then output POSSIBLE else IMPOSSIBLE.Do it using BFS,DFS.

# 11579

# Use Heron’s formula to solve it.It states that “It can be proved that to form the largest area triangle we only need to consider three consecutive sides of sorted dataset”.So solve it without DP.Use quicksort to sort the data.Here is the procedure how to sort flaoting point value.

# int comp(const void \*a,const void \*b)

# {long double \*x=(long double\*)a;

# long double \*y=(long double\*)b;

# if(\*x<\*y)

# return -1;

# else if(\*x>\*y)

# return 1;

# return 0;

# }

# 11586

# Easy adhoc problem.Just work few miniutes on it you can find out the logic.

# Test your code with following sample input:

# 4

# FF

# MM

# FF FM

# FF MM

# Output:

# NO LOOP

# NO LOOP

# NO LOOP

# LOOP

# 11498

# Very simple problem. Your task is to find out for a given co-ordinate which quadrant it possesses. But traps of this problem is that the origin is not fixed to 0 0.But don’t go back..just think what you would do if origin (0,0)…then simulate same process for also other cases..:)

# 11470

# Simulate the problem description. Not hard simulation…just careful simulation is needed.

# 11015

# Simply apply floyed warshall to compute all pair shortest path.

# 572

# Make all the position of ‘@’ sign un-visited and other position visited. Then check from index (1 1) to all (m n) position if it is unvisited apply DFS or BFS to this position for filling all other visited position adjacent it .This algorithm is simply called flood fill algorithm.

# 231

# Longest decreasing sequence.O(n^2) algorithm of DP is enough..doesn’t need O(nlogn) algorithm.

# 558

# Bellman ford algorithm is sufficient for this problem no modification is needed.

# 11770

# This is one is most tricky problem yet I solved. Algorithm to solve it is given below:

# 1.At first count the finishing time of each vertex by DFS

# 2.Now apply DFS in descending order of finishing time and compute how many vertex is unvisited.

# Here is some critical input for this problem:

# Sample Input:

# 3

# 3 3

# 1 2

# 2 3

# 3 1

# 3 2

# 2 1

# 3 2

# 5 5

# 1 2

# 2 1

# 2 3

# 3 4

# 3 5

# Sample Output:

# Case 1: 1

# Case 2: 1

# Case 3: 1

# 11504

# I solved it using code 11770.Just you need to use here quicsort.

# 11760

# Simple grid analysis. Draw some grid and test it in your paper…hope you will get the solution. One tricky input for this problem is:

# Sample input:

# 3 5 4

# 0 0

# 2 0

# 2 2

# 2 4

# 1 3

# Sample Output:

# Case 1: Escaped again! More 2D grid problems!

# 11518

# Same concept of 11504.But more simple.No topological sort is needed.DFS traversal is enough.

# 10324

# If in any problem, you are asked to end input by reading a blank line ,you must do it as following:

# While(gets(str))

# {

# }

# One more confusing information is that check your exe file whether it produces output for all the sample input given from paste option…if it modify it using c++ cin,cout,endl.

# 11158

This is not very simple ,not very hard. Finding out logic for this problem is very easy…but hard to implement(Yet I have not learned about Greedy algo).At first sort the data. Then pick up upper and lower value and find absolute difference between this value and add it with final sum. Then pick one by one value from Index 2 to N-1 and check it whether for

Tricky Input:

2

5 1 400 90 10 30

6 6 7 2 9 1 8

Output:

Case 1: 928

Case 2: 29

# 10803

Read carefully the problem.It wants “Send Kurdy” to be printed if any of city is unreachable and here unreachable does not necessarily mean that the cities should have any connection.If distance between two city is greater than 10,then the “Send Kurdy”, rule follows.This is the first geometrical problem which I solved using Graph(Floyd Warshall).

# 10596

Well this is a confusing problem…I don’t know why some suggest to deign this problem as undirected graph. I solved it using general concept. My algorithm is:

1.Check for every vertex that is included in given road whether total degree is even.

2.Check from any node that is in road construction whether all other node is reachable.

Input:

3 2

0 1

1 0

3 0

Output:

Possible

Not Possible

# 10191

You CAN'T assume, however, that the input will be in any specific order. That is it is not necessary the time you are given will be in sorted order. Be careful about input. Use sscanf() to separate digits from string.

# 10227

Some critical input for this code ..

3

2 1

1 1

2 1

1 1

2 1

3 4

3 4

1 2

2 2

Output:

1

1

2

# 482

To parse the input of this problem refer to my article “Bug In Parsing Input”. I used sort() function of STL defined in <algorihm>.

11450

This problem is a kinda 0-1Knapsack like problem.But here No weight is given rather only price is

given.I used price both as weight and price.Another varaint from 0-1knapsack is here you'll be given

products as bundle.Under which several model of same product is given.So,A bundle means a single product.

Other variant are,unlike Knapsack you must take at least one element form each bundle to maximze the item taken which

won't exceed M.To do this I omiited p[i-1][j] of original knapsack.Because it means skipping cuurent item.But

here u'll need at least one item from each bundle.And other marginal change also brought.Look at the code below

to detect this changes.