Dhaka Regional 2011: How to solve

Problem A – Binary Matrix by Arifuzzaman Arif

This problem can be solved in two steps, in first step make all rows have equal number of ones and in second step make all columns have equal number of ones. It can be proved that If both possible the summation is the result. Well now for solving one only for rows, the problem can be considered as a group of n persons in a circle and each of them have zero or more coins. They can pass some coins to their left and right. By passing coins make sure that every person has equal number of coins. What is the minimum number of passing required to do that? This problem can be converted to a median finding problem. A nice analysis of this can be found here http://one-problem-a-day.blogspot.com/2011/08/uva-11300-spreading-wealth.html.

Problem B – Candles by Shahriar Manzoor

You have to pre-calculate all the answers before starting taking input. You can use bitmask to store pre-calculated values.

Problem D – Game of Connect by Abdullah al Mahmud

The problem is a restricted instance of Shannon Switching Game. The second player will have a winning strategy if and only if the graph contains 2 disjoint spanning tree. Checking if the a given graph has a two disjoint spanning tree can be tricky. The standard approach can be to maintain disjoint forest F1 and F2. If an edge does not create any cycle in any of these forests then add the edge to that forest. Otherwise suppose if the edge create a cycle in F1, remove one of the edges from the cycle and try to add it to F2. This edge may create a cycle in F2, remove one of the edges from the cycle and add it to F1. Repeat this operation till the added edge does not create any cycle in the current forest. Let us try to add the edges through augmenting path like this. Finally at the end check if both of these F1 and F2 are trees.

Problem E – Guards by Jane Alam Jan

The solution for this problem is a bit tricky, let dp(n, k) be our solution, where we have n * n board and k is the number of cycles..

dp(n, k) {

// let in the first column we place two stars,

```
+-+-+-+-+
|*|  |  |  |
+-+-+-+-+
|*|  |  |  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
```

So, in the first column we can chose two rows in n*(n-1)/2 ways, Now since we need two stars in the rows also, so, let we place two more stars in those rows, We will place them in different columns, otherwise a cycle will be formed.

```
+-+-+-+-+
|*|*|  |  |
+-+-+-+-+
|*|  |*|  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
```

So, we can think of another dp function where we have r rows, c columns and in two rows, we have two stars.. let this be dp2(r, c, k).

Now we can chose these two columns in (n - 1) * (n - 2) ways.

so, dp1(n, k) = n * (n - 1) / 2 * (n-1)*(n-2) * dp2(n - 1, n - 2, k)
+ dp1(n - 2, k - 1) * n * (n - 1) / 2 * (n - 1) /* this part because we have places the two stars in the same column and they do form a cycle so, we can use dp1, as 2 rows and two columns are not needed and a cycle has been formed.*/

Now lets design dp2(r, c, k) {

Since in this part two rows and two columns are already filled with 2 stars, And let c < r (as we can think), so, if we form a cycle, we can place it in c ways. And if we place two stars in different columns then it can be done in c * (c-1) ways, so

```
+-+-+-+
|*|  |  | <- these row is already filled
+-+-+-+
|  |*|  | <- these row is already filled as we called dp2
+-+-+-+       ( n - 1, n - 2, k )
|  |  |  |
+-+-+-+
|  |  |  |
+-+-+-+
```

dp2(r, c, k) = dp1(c - 1, k) * c + dp2(c - 1, k) * c * (c - 1);

As we can see r = c + 1, and we can maintain this throughout the process, so, dp2(n, k) is enough.

This complexity is O(n*k).

To be more exact, we maintain one dp1 who takes an n * n board which is fully empty. And dp2 takes an m * n board where the shape is

```
+-+-+-+-+
|  |*|  |  |
+-+-+-+-+
|*|  |  |  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
|  |  |  |  |
+-+-+-+-+
```

From these two parts, we can always maintain these two kinds of small boards.

Problem F – Packing for Holiday by Arifuzzaman Arif

This is the give away problem. You need to check whether length of any side of the box exceeds 20. In case you did not solve this problem, try to practice more before doing onsite contests again.

Problem G – Pair of Toucing... By Towhidul Islam Talukder

Suppose, two circles have centres (x_1, y_1) and $(x_2\ y_2)$ and radii r_1 and r_2 respectively. Now if $x_1 = x_2$ or $y_1 = y_2$ then the cases are trivial. To calculate the number of circles where $x_1 = x_2$ we can choose a radius for the first circle r_1 and then centre it in (r_1, r_1). Then we can place circle of radius 1 to $\min\left(r_1, \frac{W-2r_1}{2}\right)$ adjacent to it. Now we may consider a bounding box of size $2r_1$ by W. In this bounding box the number of ways can be found using an arithmetic series. Then we can find the total number of ways the bounding box can be placed in the rectangle and then multiply. We loop through all possible r_1 and can calculate all the pairs that have $x_1 = x_2$. Similarly we can calculate the number of pairs where $y_1 = y_2$. These procedures have the complexities of O(W) and O(H) respectively. However, a O(n^2) loop over all possible r_1 and r_2 was also allowed.

Now we need to calculate the number of circles where $x_1 \neq x_2$ and $y_1 \neq y_2$. Note that for this to be true, the x difference, the y difference and the Euclidian distance between two centres will form a Pythagorean triple. Let (a, b, c) is a Pythagorean triple. Then we can draw (c – 1) pairs of circles in a bounding box of size (a + c) by (b + c). This cause some over counting as in the corner some cases with big circles are invalid. These cases can be detected and removed. We can calculate total pairs of circles associated to this triple by calculating the number of ways the bounding box can be placed on the rectangle (without rotation) and then multiplying it. To calculate the number of all possible pairs we can iterate through all the Pythagorean triples in that range. Another option is for each Pythagorean triple loop over c to find a pair and then find the number of ways they can be placed in the rectangle. This solution is slower but was allowed.

To avail we can pre-generate all the Pythagorean triple in the maximum input range. An $O(n^2)$ loop should suffice.

Problem H – Treasure ... By Shahriar Manzoor

Suppose the center of gravity of the plate is G and center of gravity of four people is O. To make the plate stable O must be shifted to G. So the minimum movement of O is OG. This can be achieved by translating all the persons by the amount OG, in the direction parallel to OG (It can be proved that this is minimum). So the total minimum movement is 4OG. But while doing so some person may go out of the plate, which is not allowed. In that case that person will stop on the boundary and another person, who has provision to make longer move, should cover that move. This algorithm may sometimes fail if the plate is not rectangular shaped.

Problem I – Truchet Tiling ... By Monirul Hasan

Notice that we can analyze the individual cells in the grid to see which part of the cell gets hit and add up the corresponding area. The area calculation for each cell requires trivial geometry. For Motif 0, top left or top right corner hits means 1/4π area gets selected, top right, bottom left or center hit means 4-1/2π area addition. Area calculations for Mofit 1 is similar. Now given an initial position to compute total contagious area we can start at the initial position and then visit the connected 8 cells as needed; the trick is to keep track of how we are entering that particular cell (from bottom left, bottom right, top left,top right or the middle). Thus this problem can be transformed into a graph search problem where the state information holds cell number and position it was visited from. The cases where we hit a curve is trivial to detect because they can only happen if either (not both) of the row/column component of the starting position is an odd number.

Problem J – As Long as I... by Jane Alam Jan

This is a simulation problem. You can represent it as graph or solve it even without doing so.

Problem C – Cards by Sohel Hafiz

Without the joker cards, it's a standard DP problem with state space [13][13][13][13]. The joker cards make the problem a little trickier. To handle them, we have to increase the dimension to [13][13][13][13][2]. In every recursive call, if you get a regular card, you simply calculate the expected moves by increasing the corresponding a,b,c,d by one in [a][b][c][d]. If you get a joker card, try every possibility and see which one yields the minimum result.