**Week-1**

Q1. Hello World

Just print the name

Q2. Relativity

Find E=cm2 use long

Q3. Chess Tournament

Sort values inside individual row and then sort each row according to the skill levels.

Q4. Scrambled Chatlog

Split the string with # and find the substring according to the number given and then split it after which concatenate it.

Q5. Watson

Replace each word with the symbols, use stack based expression evaluation and give priority as the same.

**Week 2**

MST

Prisms

running time (jV j = n, jEj = m)

everything without PQ: O(n + m)

n insert: O(n)

m decreaseKey: O(m)

n deleteMin: O(n log n)

O(m + n logn)

Kruskal

sort O(mlogm)

2m Find O(m alpha (m,n))

N union O(n alpha (m,n))

O(mlogm)

Alpha is inverse Ackermann function which is a little slower than linear.

Q1. Pizza toppings

Create edges between topping which does not go together and check for bipartiteness.

Q2.Networking

Find MST of a graph where people are the nodes and distance are the edges.

Q3. Road destruction

Create graph with cities as nodes as edges roads with weights lane\*no of lanes

Find MST weight , subtract it from total weight .

Impossible if graph is not connected.

Q4. Technical Difficulties

Remove each edge and run kruskal. If it is not connected after that output that edge.

**Week 3**

Dijkstras Algorithm

Running time: like Prim's algorithm O(m + n log n)

Bellman Ford

per phase: O(n + m) (dense graph m = n2 ) O(n2))

total: O(n(n + m)) = O(n . m)

Q1. Hiking

Shortest Path, bi directed graph

Q2.Party Planning (Don’t Know)

Use bellman ford to find shortest path.

Q3. Snake and Ladder

Create a graph according to the snake and ladder info plus for each dice throw.

If its snake then edge to end of snake

If tis ladder edge to end of ladder plus edge to the next node.

Run dijkstras and find the num of moves

Output the minimum number of moves

Q4.SuperMarket

Create adj matrix with cities as nodes and roads as edges.

Take each supermarket . Add the additional wait time to the edges of that city.

Find shortest path to all nodes from that supermarket city .

Find shortest dist to lea and peter add them and store.

For each supermarket city do the same.

Keep on updating the min distance and print it as time format.

**Week 4**

Dinic

works always

running time: O(|V|^2 .|E|)

use BFS (which Dijkstra does for unit lengths)

Push and Relabel (Goldberg and Tarjan)

it is necessary to erase all nodes that are not reachable first, otherwise there are some nasty corner cases!

nice running time O(|V |^3)

Q1. Chocolate

Max flow with single source and single sink

Q2.Bank Robbery

Create a bi directional graph check the maxflow and if the maxflow <= num of cops then yes

Q3.Christmas Presents (Sreenath)

Max flow Bipartite Matching , if the maxflow = num of friends

Q4. Beer Pipes

Max flow, calculating the capacity properly is the catch and precision handling

Q5.Football Champion(Parvez) (Baseball elimination http://www.cs.princeton.edu/courses/archive/spr03/cos226/assignments/baseball.html)

Check first after all the games played the team can win. By comparing the maximumCurrentWin with the teams maxWin

Then take all the teams who made the cut and create a graph with a source and a sink

Source connected to the match and the match is connected to both the teams and the target connect to them.Teams are not connected

Check if the maxflow = maximumCurrentWin

Then that team can win the competition.

that is, *if* all the games can be played so that teams Y, C, B get no more than u(Y), u(C), u(B) wins correspondingly, *then* Harvard still can be number one

**Week 5**

Q1. Contact List

Tries, Make a Tries tree see if the node is full word and it has children then increment the counter.

Q2.Cable Car

Calculate distance without canyon

double distanceWithoutCanyon = lengthOfRoute / (double)(numOfPosts - 1);

if(startCanyon == endCanyon)

return distanceWithoutCanyon

Find num of posts before the start of the canyon and num of posts before the end of the canyon plus 1 to take care of the post at 0

numberOfPostsNeedingRelocation = (postJustAfterCanyon - postJustBeforeCanyon - 1)

if(numberOfPostsNeedingRelocation == 0)

return distanceWithoutCanyon;

if(startCanyon == 0 && endCanyon == lengthOfRoute)

return zero

if(startCanyon == 0)

put all the posts on the right and find the gap

if(endCanyon == lengthOfRoute)

put all posts on the left and find the gap between the posts

Else

First put all the posts on the right of the canyon and zero of the left and move each post that need recolation on to the right one by one and calculate the gap.

Return the maximum gap value

Q3.Warp speed ahead

Use BigInteger and appl the formula correctly

Q4.Game Show

Use big integer

Find the final denominator first

Find the numerator for each power and denominator of each power

Divide the factor the numerator should be multiplied by to make fraction addition possible by dividing the final denominator with the current denominator.

Multiply the factor with the numerator and then add it with the total numerator.

At the end find gcd of num and denom and print the answer by dividing num and denom with the gcd

**Week 6**

Q1. Sudoku

Used Code from internet

Q2. Story Time

Recursion

Parvez need to explain

Q5. Queens

<http://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem>

Checked the presence of a queen towards the right upper left and upper right directions as well

Use recursion

**Week 7**

Q1. Coin Change

Q2. Zombie Apocalypse

<http://www.codeproject.com/Articles/706838/Bounded-Knapsack-Algorithm>

Q3.Vaults and Vampires

<http://www.geeksforgeeks.org/dice-throw-problem/>

Q4. Packing cases

<http://www.geeksforgeeks.org/dynamic-programming-set-21-box-stacking-problem/>

Q5 Birthday Cake

<http://www.geeksforgeeks.org/dynamic-programming-set-11-egg-dropping-puzzle/>

**Week 8**

Q1.Goldmans Conjecture

<http://introcs.cs.princeton.edu/java/14array/Goldbach.java.html>

Q2.Commander in Chief

Find GCD repeatedly

Q3.N-Athlon

Chinese Remainder Theorem with BigInteger

Q4. Cookies

Find all the sums that can be obtained from the given number of cookies that each friend eats.

Then for each combination get the LCM and then print after taking mod with the given number

Q5.Soft Skills (Have to ask)

Sreenath

Lucas Theorem

nCr Mod p we split it to n0Cr0 . n1Cr1. N2Rc2 … Modp

and for each such modulo you save and apply CRT for all the modulos saved

**Week 9**

Q1.Meteorite(Need to confirm)

Using DFS add all the points to a list and see if the impact points is part of that list.

Q2.Fence Posts

Convex Hull

Q3. Burger and Chips

Circumcircle radius of the polygon formed my connecting all the burger centers.

Add the radius of one burger to it.

Q4.Fragile Letter (Parvez)

**Week 10**

Q1. Eurlers

Centroid – intersection of medians

X1+x2+x3/3, y1+y2+y3/3

Circumcenter = intersection of perpendicular bisectors

The [Cartesian coordinates](https://en.wikipedia.org/wiki/Cartesian_coordinates) of the circumcenter are

U_x = \left[(A_x^2 + A_y^2)(B_y - C_y) + (B_x^2 + B_y^2)(C_y - A_y) + (C_x^2 + C_y^2)(A_y - B_y)\right] / D,

U_y = \left[(A_x^2 + A_y^2)(C_x - B_x) + (B_x^2 + B_y^2)(A_x - C_x) + (C_x^2 + C_y^2)(B_x - A_x)\right]/ D

with

D = 2\left[A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)\right].\,

Orthocenters – intersection of altitudes

Q2. Templonian Excavation

Find Area, find closest multiple of 3. Find integer closes multiple of 3 and multiply by 3

Q3. Alien

Find Area

Q5.