Group1 :-Dynamic Programming

1. Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, … shows the first 11 ugly numbers. By convention, 1 is included. Given a number n, the task is to find n'th Ugly number.

2. Given a M x N matrix where each cell has a cost associated with it, find the minimum cost to reach last cell (M-1, N-1) of the matrix from its first cell (0, 0). We can only move one unit right or one unit down from any cell. i.e. from cell (i, j), we can move to (i, j+1) or (i+1, j). Find the minimum cost to reach last cell of the matrix from its first cell.
3. Given an array A, maximize value of the expression (A[s] - A[r] + A[q] - A[p]) where p, q, r and s are indexes of the input array and s > r > q > p. We can use Dynamic Programming to solve this problem.
   For example:
   Input: A[]=[3,9,10,1,30,40]
   Output: 46
   (40-1+10-3) will result in maximum value.
4. Given an array of integer numbers, we need to sort this array in a minimum number of steps where in one step we can insert any array element from its position to any other position.
   Example:

   Input : arr[] = [2, 3, 5, 1, 4, 7, 6]

   Output : 3

   We can sort above array in 3 insertion

   steps as shown below,

   1 before array value 2

   4 before array value 5

   6 before array value 7

5. Given a text and a wildcard pattern, implement wildcard pattern matching algorithm that finds if wildcard pattern is matched with text. The matching should cover the entire text (not partial text).

   The wildcard pattern can include the characters '?' and '*'
   '?' – matches any single character
   '*' – Matches any sequence of characters (including the empty sequence)

   For example,

   Text = "baaabab",

   Pattern = "*****ba*****ab", output : true

Pattern = "ba*a?", output : true

6. Consider a row of n coins of values v1 . . . vn, where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Determine the maximum possible amount of money we can definitely win if we move first.

   8, 15, 3, 7 : The user collects maximum value as 22(7 + 15)

   Does choosing the best at each move give an optimal solution?
   …….User chooses 8.
   …….Opponent chooses 15.
   …….User chooses 7.
   …….Opponent chooses 3.
   Total value collected by user is 15(8 + 7)

7. We know that prime numbers are positive integers that have exactly two distinct positive divisors. Similarly, we'll call a positive integer t T-prime, if t has exactly three distinct positive divisors. You are given an array of n positive integers. For each of them determine whether it is T-prime or not.

8. There are n houses built in a line, each of which contain some value in it. A thief is going to steal in these houses. But he cannot steal in two adjacent houses. What is maximum value he can steal?

9. There are 'p' balls of type P, 'q' balls of type Q and 'r' balls of type R. Using the balls we want to create a straight line such that no two balls of same type are adjacent.

   Example:

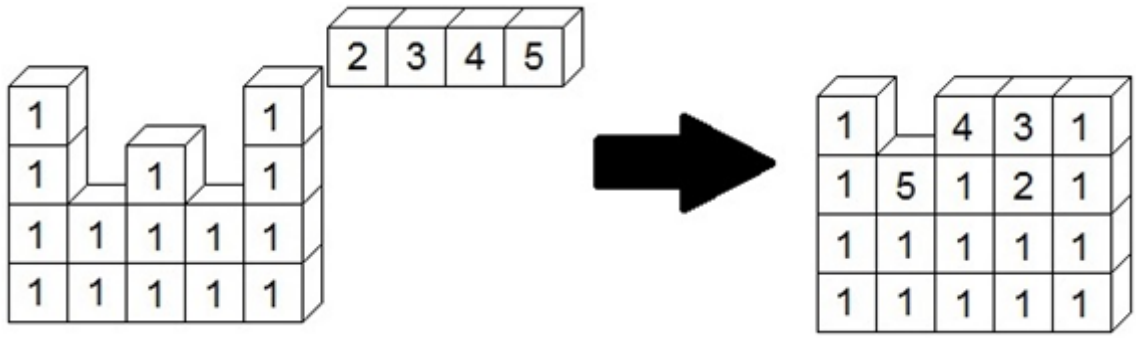
   Input : p = 2, q = 1, r = 1

   Output : 6

   There are only six arrangements PQRP, QPRP,

   PRQP, RPQP, PRPQ and PQPR

10. The Company Blocks Regular Inventing Usefulness of Something, better known as Brisa, build blocks, always the same size . One detail that stands out is the manner in which the blocks are stored in stock, after manufactured . They are formed by a row of cells . Withdrawal of a stock box is somewhat cluttered when , for choosing a cell at random and cut up some top block it. However , the storage medium is somewhat interesting: a conveyor located at the top of the stack straight rightmost stock is used . With this, it forms a queue with the new blocks . The belt right wheel to the left. So there is a vacant space in one of two cells, the block will be inserted in it, if there is not, it progresses to the following cells. Below is an example of insert blocks .



Input

There will be several test cases . Each test case have three integers , M , P and F , indicating the rightmost stack height , the number of stacks of blocks and the size of the row of blocks to be inserted . Following this, M lines are read with P values , with values 1 , which is represented block, and 0 , representing which does not block. Next, a line is read with F values representing the queue with the new blocks . The last test case is represented by three zeros , and should not be processed.

Output

For each test case , print the cells after the addition of new blocks. In some cases, a row of new blocks is more than sufficient for all the cells remain the same size. In this case , disregard the blocks that are left in the queue.

Input Sample	Output Sample
4 5 4 1 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 2 3 4 5 5 3 6 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 4 5 6 7 8 9 0 0 0	1 0 4 3 1 1 5 1 2 1 1 1 1 1 1 1 1 1 1 1 1 8 1 1 7 1 1 6 1 1 5 1 1 4 1

Tasks: Your specific tasks are as follows:
(1) Provide a detailed description of the working of the algorithm along with an example.
(2) Provide a pseudo code for the algorithm.
(3) Analyze the theoretical run-time complexity of the algorithm and show that it is $\Theta(n \log n)$.

(4) Analyze the theoretical space complexity of the algorithm.
(5) Discuss the correctness of your algorithm
(6) Implement the algorithm

11. Counting the Number of Inversions in an Array

Your **objective** in this project is to develop a Θ(nlogn) algorithm to determine the inversion count of an array.
The number of inversions in an array is a measure of how far is the array from being sorted. For an array A[0, ..., n-1], we say there is an inversion involving two distinct indices i and j (where $0 \leq i \leq n-1$ and $0 \leq j \leq n-1$) if $i < j$, but $A[i] > A[j]$. The number of such pairs of indices i and j (wherein $i < j$ and $A[i] > A[j]$) in an array A is called the inversion count of the array.
For example: the following array has 4 inversions

0	1	2	3	4
10	1	4	10	5

Index Pairs i, j	Values A[i], A[j]	Inversion Yes or No
0, 1	10, 1	Yes
0, 2	10, 4	Yes
0, 3	10, 10	No
0, 4	10, 5	Yes
1, 2	1, 4	No
1, 3	1, 10	No
1, 4	1, 5	No
2, 3	4, 10	No
2, 4	4, 5	No
3, 4	10, 5	Yes

Note that if for any two distinct indices i and j, if A[i] = A[j], then there is NO inversion involving the indices i and j. In the example shown here, there is no inversion involving indices 0 and 3, as both A[0] and A[3] are 10. We say there is an inversion for two distinct indices i and j only if $i < j$ and $A[i] > A[j]$.

Tasks: Your specific tasks are as follows:
(1) Provide a detailed description of the working of the algorithm along with an example.
(2) Provide a pseudo code for the algorithm.
(3) Analyze the theoretical run-time complexity of the algorithm and show that it is Θ(nlogn).
(4) Analyze the theoretical space complexity of the algorithm.
(5) Discuss the correctness of your algorithm
(6) Implement the algorithm
(7) Run simulations of your algorithm for array sizes of 10, 100, 1000 and 10000. You could fill the contents of the arrays with random integers ranging from 1 to 500. For each array size, run

the simulations for 50 runs and average the results for the inversion and the execution time of the algorithm. Plot the results for the array size (n) vs. {average execution time and the theoretical run-time complexity that is supposed to be nlogn} and also plot the results for the array size vs. average inversion count.

Group2:-Trees

1. Program for finding the largest value in a Binary Search Tree using Inorder traversal.
2. Program to construct a binary tree.
3. Program to calculate number of leaf nodes in a tree.
4. Program print height and depth of given binary tree.
5. Write program to find the total number of non leaf nodes present in a tree using recursion. Create a recursive function which takes in root of the tree as input and returns count of number of non leaf nodes present in a tree.
6. Program to find the Sum of all Nodes in a Tree such that any node is sum of values at left and right sub tree.
7. Program to print all the Paths from the Root to the Leaf in a Tree.
8. Program to find nodes which are at maximum distance in binary tree.
9. Given two binary trees, check if the first tree is subtree of the second one. A subtree of a tree T is a tree S consisting of a node in T and all of its descendants in T.
10. **Kitty's Calculations on a Tree.** Kitty has a tree, T, consisting of n nodes where each node is uniquely labeled from 1 to n . Her friend Alex gave her q sets, where each set contains k distinct nodes. Kitty needs to calculate the following expression on each set:

$$\left(\sum_{\{u,v\}} u \cdot v \cdot dist(u,v)\right) \bmod (10^9+7)$$

where:

- {u,v} denotes an unordered pair of nodes belonging to the set.
- Dist{u,v} denotes the number of edges on the unique path between nodes u and v.

Given T,q and k sets of distinct nodes, can you help her calculate the expression for each set?

For each set of nodes, print the value of the expression modulo $10^9 + 7$ on a new line.

Input Format

The first line contains two space-separated integers describing the respective values of n (the number of nodes in tree T) and q (the number of sets).

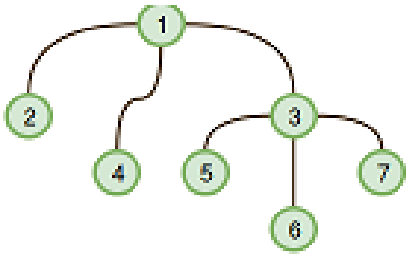
Each of the n-1 subsequent lines contains two space-separated integers, a and b , describing

an *undirected* edge between nodes a and b .

The 2.q subsequent lines define each set over two lines in the following format:

1. The first line contains an integer, k , denoting the size of the set.
2. The second line contains k space-separated integers describing the set's elements.

Consider tree T,



We perform the following calculations for $q = 3$ sets:

- Set 0: Given set $\{2, 4\}$, the only pair we can form is $(u, v) = (2, 4)$, where $dist(2, 4) = 2$. We then calculate the following answer and print it on a new line:

$$(2 \cdot 4 \cdot dist(2, 4)) \bmod (10^9 + 7)$$
$$\Rightarrow (2 \cdot 4 \cdot 2) \bmod (10^9 + 7)$$
$$\Rightarrow 16$$

- Set 1: Given set $\{5\}$, we cannot form any pairs because we don't have at least two elements. Thus, we print 0 on a new line.
- Set 2: Given set $\{2, 4, 5\}$, we can form the pairs $(2, 4)$, $(2, 5)$, and $(4, 5)$. We then calculate the following answer and print it on a new line:

$$(2 \cdot 4 \cdot dist(2, 4) + 2 \cdot 5 \cdot dist(2, 5) + 4 \cdot 5 \cdot dist(4, 5)) \bmod (10^9 + 7)$$
$$\Rightarrow (2 \cdot 4 \cdot 2 + 2 \cdot 5 \cdot 3 + 4 \cdot 5 \cdot 3) \bmod (10^9 + 7)$$
$$\Rightarrow 106$$

Group3:-String Algorithm

1. Java program to check if two given strings are anagrams of Each other. Two strings are anagrams if they are written using the same exact letters, ignoring space, punctuation, and capitalization. Each letter should have the same count in both strings. For example, the Army and Mary are an anagram of each other.
2. Find the first non-repeated (unique) character in a given string. for Example, if given String is "Morning" then it should print "M".
3. Write a program to print all duplicate character and their count. For example, if given String is "Programming" then your program should print

   g : 2
   r : 2
   m : 2

4. Sherlock considers a string to be *valid* if all characters of the string appear the same number of times. It is also *valid* if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times. Given a string , determine if it is *valid*. If so, return YES, otherwise return NO.
   For example, if s=abc , it is a valid string because frequencies are {a:1,b:1,c:1} . So is s=abcc because we can remove one c and have 1 of each character in the remaining string. If s=abccc however, the string is not *valid* as we can only remove 1 occurrence of c . That would leave character frequencies of {a:1,b:1,c:2}.
5. Write a program to check if two given String s1 and s2 are rotations of another. For example if s1 = "IndiaUSAEngland" and s2= "USAEnglandIndia" then your program should return true but if s2="IndiaEnglandUSA" then it should return false.
6. Given three strings, return true if third string is interleaving of first and second string. i.e., it is formed from all characters of first and second string and order of characters is preserved.
   For example, ACDB is interleaving of AB and CD

7. A gene is represented as a string of length n (where n is divisible by 4), composed of the letters A,C ,T , and G. It is considered to be *steady* if each of the four letters occurs exactly n/4 times. For example, GACT and AAGTGCCT are both steady genes.
   Bear Limak is a famous biotechnology scientist who specializes in modifying bear DNA to make it steady. Right now, he is examining a gene represented as a string ‘gene’. It is not necessarily steady. Fortunately, Limak can choose one (maybe empty) substring of ‘gene’ and replace it with any string of the same length.
   Modifying a large substring of bear genes can be dangerous. Given a string ‘gene’, can you help Limak find the length of the smallest possible substring that he can replace to make ‘gene’ a steady gene?
   As an example, consider gene=ACTGAAAG . The substring AA just before or after G can be replaced with CT or TC . One selection would create ACTGACTG .

Group-6 Recursion

1. Write a recursive function that computes the sum of all numbers from 1 to n, where n is given as parameter.

```
//return the sum 1+ 2+ 3+ ...+ n

int sum(int n)
```

2. Write a recursive function that finds and returns the minimum element in an array, where the array and its size are given as parameters.

```
//return the minimum element in a[]

intfindmin(int a[], int n)
```

3. Write a recursive function that computes and returns the sum of all elements in an array, where the array and its size are given as parameters.

```
//return the sum of all elements in a[]

intfindsum(int a[], int n)
```

4. Write a recursive function that determines whether an array is a palindrome, where the array and its size are given as parameters.

```
//returns 1 if a[] is a palindrome, 0 otherwise

intispalindrome(char a[], int n)
```

5. Write a recursive function that searches for a target in a sorted array using binay search, where the array, its size and the target are given as parameters.

6. Write a function for mutliply(a,b), where a and b are both positive integers, but you can only use the + or − operators. 4. In the lecture, we discussed a method to raise a double to an integer power. In this question, write a recursive function that allows raising to a negative integer power as well.

7. Write a recursive function to reverse a string. Write a recursive function to reverse the words in a string, i.e., ”cat is running” becomes ”running is cat”.

8. [Medium hard] A word is considered elfish if it contains the letters: e, l, and f in it, in any order. For example, we would say that the following words are elfish: whiteleaf, tasteful, unfriendly, and waffles, because they each contain those letters. • Write a predicate function called elfish? that, given a word, tells us if that word is elfish or not. • Write a more generalized predicate function called x-ish? that, given two words, returns true if all the letters of the first word are contained in the second.

9 [Hard example] Coin game: Alice and Bob are playing a game using a bunch of coins. The players pick several coins out of the bunch in turn. Each time a player is allowed to pick 1, 2 or 4 coins, and the player that gets the last coin is the winner. Assume that both players are very smart and he/she will try his/her best to work out a strategy to win the game. For example, if there are 2coins and Alice is the first player to pick, she will definitely pick 2 coins and win. If there are 3 coins and Alice is still the first player to pick, no matter she picks 1 or 2 coins, Bob will get the last coin and win the game. Given the number of coins and the order of players (which means the first and the second players to pick the coins), you are required to write a program to calculate the winner of the game, and calculate how many different strategies there are for he/she to win the game. You should use recursion to solve the problem, and the parameters are read from the command line. You can assume that there are no more than 30 coins.

Here are some sample runs of the program: ./pickcoin 1 alice bob alice 1 ./pickcoin 2 bob alice bob 1 ./pickcoin 3 alice bob bob 2 ./pickcoin 10 alice bob alice 22 ./pickcoin 25 alice bob alice 3344 ./pickcoin 30 alice bob

Group-Divide & Conquer

Exercise in Class

a) **Merge Sort**
b) **Quick Sort**
c) **Binary search**
d) **Max-Min using Divide & Conquer**
e) **Longest Common Prefix problem**

Practice Exercise for Group-

Problem 1. Count the number of inversions (out of order pairs) in an array. Let the elements be x1,...,xn. There is an inversion if xi >xj and i< j.

We can divide and conquer like in merge sort. The idea:

1. Divide array into left and right.

2. Continue step 1 until we get to one element subarrays. Return 0 inversions when we do.

3. Calculate inversions between the two by combining left and right.

(a) Let the current index in left be j and the index in right be k. (b) If right[k] < left[j], everything between the two is out of order.

4. Inversions = invs in left + invs in right + invs between left and right.

5. Return inversions.

This procedure runs in O(nlogn) time. There are logn split / merge steps, processing n elements each.

Problem 2. Suppose you are investing. You want to buy high, then sell low. You have an array x of integers representing future prices, and can make one buy and one sell. What is the most you can make?

This is another divide and conquer problem.

1. Divide array into left and right.

2. Return 0 when we get to arrays of size 1.

3. When we merge, the maximum profit is the maximum of the profit in left, the profit in right, and the maximum profit between the two.

4. The max profit between is the difference of the maximum in right and the minimum in left.

5. Return maximum profit.

Runtime is the same as the previous problem, O(nlogn).

Problem 3 Given a n by n board where n is of form 2k where k >= 1 (Basically n is a power of 2 with minimum value as 2). The board has one missing cell (of size 1 x 1). Fill the board using L shaped tiles. A L shaped tile is a 2 x 2 square with one cell of size 1×1 missing.

Problem 4: We are given an array of n points in the plane, and the problem is to find out the closest pair of points in the array. This problem arises in a number of applications. For example, in air-traffic control, you may want to monitor planes that come too close together, since this may indicate a possible collision. Recall the following formula for distance between two points p and q.

$$\|pq\| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

Use Divide & Conquer Approach for the same.

Problem 5: Solve Stair case problem using Divide & conquer approach.

Problem 6 : Solve Tower of Hanoi Problem using Divide & conquer approach.

Problem 7: Solve Convex Hull Problem using Divide & Conquer approach.

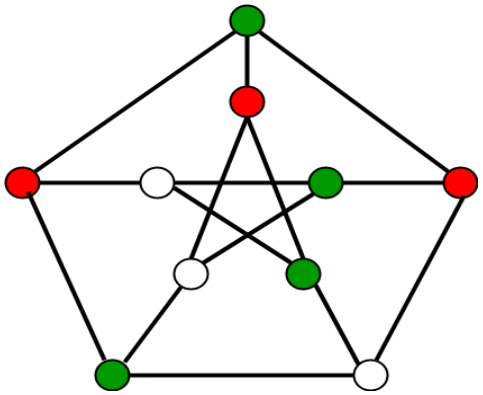
Group :-Graph Algorithms

Exercises in Class

a) Representation of Graph
b) BFS for a Graph
c) DFS for a Graph
d) Detect cycle in a Graph

Practice Questions

1. Given a tree represented as undirected graph. Count the number of nodes at given level l. It may be assumed that vertex 0 is root of the tree.
2. Count the total number of ways or paths that exist between two vertices in a directed graph. T.
3. Given a directed graph, a source vertex 'src' and a destination vertex 'dst', print all paths from given 'src' to 'dst'.
4. You are given a undirected graph G(V, E) with N vertices and M edges. We need to find the minimum number of edges between a given pair of vertices (u, v).
5. Given a directed graph, check whether the graph contains a cycle or not. Your function should return true if the given graph contains at least one cycle, else return false
6. In a given graph with a certain number of vertices, check if the vertices can be coloured in such a way that no two adjacent vertices have a similar colour. Also, check if such a colouring can happen with utmost 'm' colours. The value of the maximum number of colours that can be used is provided by the user as input. Consider the following graph as an example which can be filled with utmost 3 colors.

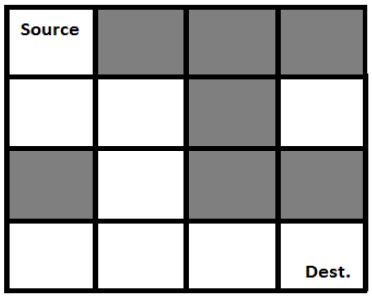


7. A Maze is given as N*N binary matrix of blocks where source block is the upper left most block i.e., maze[0][0] and destination block is lower rightmost block i.e., maze[N-1][N-1]. A rat starts from source and has to reach the destination. The rat can move only in two directions: forward and down.

In the maze matrix, 0 means the block is a dead end and 1 means the block can be used in the path from source to destination. Note that this is a simple version of the typical Maze problem. For example, a more complex version can be that the rat can move in 4 directions and a more complex version can be with a limited number of moves.

Following is an example maze.

Gray blocks are dead ends (value = 0).



Following is binary matrix representation of the above maze.

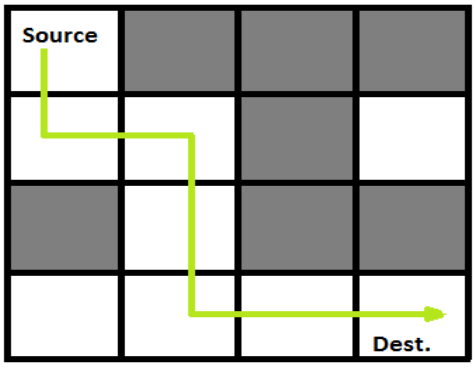
{1, 0, 0, 0}

{1, 1, 0, 1}

{0, 1, 0, 0}

{1, 1, 1, 1}

Following is a maze with highlighted solution path.



Following is the solution matrix (output of program) for the above input matrx.

{1, 0, 0, 0}

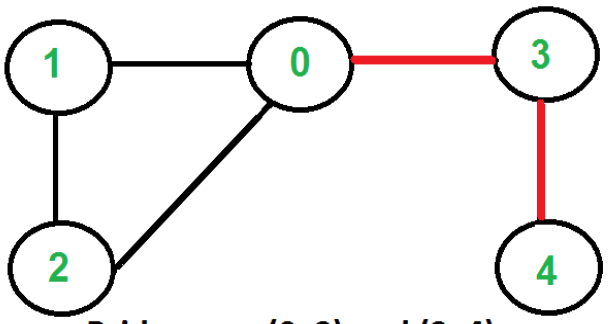
{1, 1, 0, 0}

{0, 1, 0, 0}

{0, 1, 1, 1}

All enteries in solution path are marked as 1.

8. Check if removing a given edge disconnects a graph. Given an undirected graph and an edge, the task is to find if the given edge is a bridge in graph, i.e., removing the edge disconnects the graph.



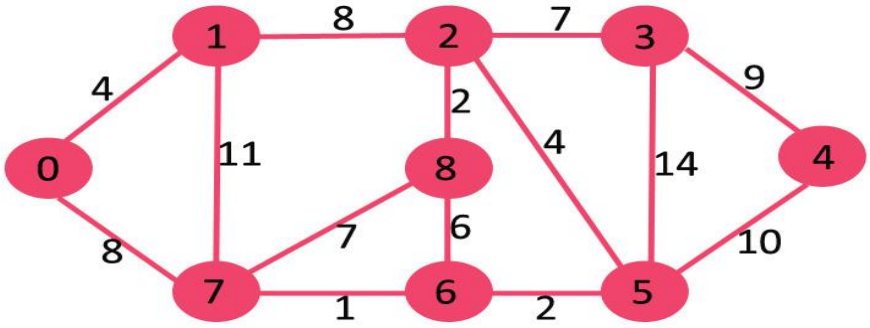
Bridges are (0, 3) and (3, 4)

Greedy algorithm

1. Implement kruskal's Minimum spanning tree algorithm.

   Below are the steps for finding MST using Kruskal's algorithm

1. *Sort all the edges in non-decreasing order of their weight.*
2. *Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.*
3. *Repeat step#2 until there are (V-1) edges in the spanning tree.*



2. Prefix Codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream.

   Let us understand prefix codes with a counter example. Let there be four characters a, b, c and d, and their corresponding variable length codes be 00, 01, 0 and 1. This coding leads to ambiguity because code assigned to c is the prefix of codes assigned to a and b. If the compressed bit stream is 0001, the de-compressed output may be "cccd" or "ccb" or "acd" or "ab".

   Implement Huffman tree and evaluate the frequency.

3. Consider the given graph, Implement Knapsack algorithm in java, where m=12, P= profits, W=weights. Analyze the complexity of your algorithm.

P	18	25	27	10	15
W	3	5	4	3	6

4. **Greedy algorithm for calculating student exam schedule**. The basic idea is to find the exam schedule which uses the fewest number of slots for a set of students. Students take a

selection of different modules each and should be able to attend all of their exams without any clashes.

Here's the input data:

John takes Modules A, G, F and C
Ben takes Modules E, F, B, and A
Clare takes Modules D, A, G, and E

And this should result in

Time slot 1: Modules D and F
Time slot 2: Modules B and G
Time slot 3: Modules C and E
Time slot 4: Module A

5. Given a set of tasks with deadlines and total profit earned on completion of a task, find maximum profit earned by executing the tasks within the specified deadlines. Assume any task will take one unit of time to execute and any task cant execute beyond its deadline. Also, only one task can be executed at a time.

   Example:

Input: Four Jobs with following deadlines and profits

JobID Deadline Profit

a 4 20

b 1 10

c 1 40

d 1 30

Output: Following is maximum profit sequence of jobs

c, a

6. You are given a string of 2N characters consisting of N '[' brackets and N ']' brackets. A string is considered balanced if it can be represented in the for S2[S1] where S1 and S2 are balanced strings. We can make an unbalanced string balanced by swapping adjacent characters. Calculate the minimum number of swaps necessary to make a string balanced.

Examples:

Input : []][][

Output : 2

First swap: Position 3 and 4

[][]][

Second swap: Position 5 and 6

[][][]

Input : [[][]]

Output : 0

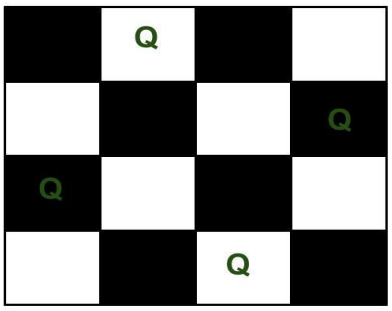
String is already balanced.

Backtracking Algorithms

1. Given a String, print all the permutations of it.

Example:
Input : abc
Output: abc acb bac bca cba cab

2. The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other. For example, following is a solution for 4 Queen problem.



3. Hamiltonian Path in an undirected graph is a path that visits each vertex exactly once. A Hamiltonian cycle (or Hamiltonian circuit) is a Hamiltonian Path such that there is an edge (in the graph) from the last vertex to the first vertex of the Hamiltonian Path. Determine whether a given graph contains Hamiltonian Cycle or not. If it contains, then prints the path. Following are the input and output of the required function.

Input:
A 2D array graph[V][V] where V is the number of vertices in graph and graph[V][V] is adjacency matrix representation of the graph. A value graph[i][j] is 1 if there is a direct edge from i to j, otherwise graph[i][j] is 0.
Output:
An array path[V] that should contain the Hamiltonian Path. path[i] should represent the ith vertex in the Hamiltonian Path. The code should also return false if there is no Hamiltonian Cycle in the graph.
For example, a Hamiltonian Cycle in the following graph is {0, 1, 2, 4, 3, 0}.

```
(0)-(1)-(2)
 |  / \  |
```

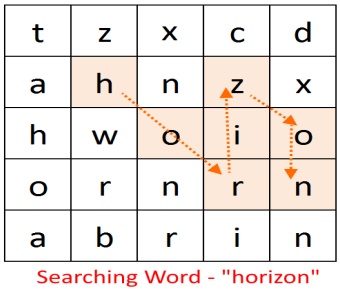
```
| /  \ |
| /    \ |
(3)-----(4)
```

4. Given an integer array of N elements, the task is to divide this array into K non-empty subsets such that the sum of elements in every subset is same. All elements of this array should be part of exactly one partition.
   Examples:

```
Input : arr = [2, 1, 4, 5, 6], K = 3
Output : Yes
we can divide above array into 3 parts with equal
sum as [[2, 4], [1, 5], [6]]

Input  : arr = [2, 1, 5, 5, 6], K = 3
Output : No
It is not possible to divide above array into 3
parts with equal sum
```

5. Given a 2D matrix of characters. Check whether the word exist in the matrix or not. If it exists then print its path. All movements are allowed (right, left, up, down and diagonally).
   For example:



Searching Word - "horizon"

6. Given a string, find out if string follows a given pattern or not without using any regular expressions.

Examples:

Input:
string - GraphTreesGraph
pattern - aba
Output:
a->Graph
b->Trees

Input:
string - GeeksforGeeks
pattern - GG
Output:
No solution exists