Google onsite interview questions

Min cost to make string palindrome:

Given a string S and a cost matrix C.

C[i][j] denotes the cost to convert character i to character j.

Goal is to convert the string into a palindromic string. In one operation you can choose a character of string and convert that character to any other character. You can do this operation any number of times. The cost to convert one character to another character is determined by the cost matrix. Find the minimum cost to convert a given string to a palindrome.

Largest island in a matrix:

Main. Followup or Followup.

Fair log pickups:

A log is defined as:

```
class Log {
      String text;
      String serverId;
}
```

You are given a list of logs and a number k. You need to pick up k logs in total from the servers. But while picking one must ensure that the pickup strategy is as fair as possible. By fair it means that it

should not happen that all the logs are picked up from the same machine. Return list of log files. Implement the following method:

```
List<Log> optimalLogPickup(List<Log> logs, int k) {

}
```

Eg:

```
logs = [{"hello", "server#1"}, {"world", "server#1"}, {"Rishika Sinha", "server#2"}, {"best PM", "server#2"}], k = 2
```

possible output:
logs = [{"hello", "server#1"}, {"Rishika Sinha", "server#2"}]
incorrect output:
logs = [{"hello", "server#1"}, {"world", "server#1"}] // Since I am picking both the logs from the same server (server#1) while being unfair to server#2.

Connected queries

There is a running stream of queries. The queries will be of the following form:

1. connect x and y (output: no output)
2. are x and y directly connected (output: true/false)
3. are x and y transitively connected (output: true/false)

Employee awareness

You are given an org hierarchy in following form:

```
{
      "employee_id: "EMP#12",
      "boss_id": "EMP#34"
}
```

This means employee_id reports to boss_id and time it takes for the boss to convey some information to the employee is 1.

There is also a CEO of the organization who does not have a boss_id. Find the maximum time it will take for the CEO to convey information to all of the employees.

Followup:

You are given an org hierarchy in following form:

```
{
      "employee_id: "EMP#12",
      "boss_id": "EMP#34",
      "convey_time": 12312
}
```

This means employee_id reports to boss_id and time it takes for the boss to convey some information to the employee is convey_time.

There is also a CEO of the organization who does not have a boss_id. Find the maximum time it will take for the CEO to convey information to all of the employees.

Topological sorting

Topological sorting :)

Splitting the string

Given a string. You have to split the string based on space. Eg:

```
str = "I love India"
output: ["I", "love", "India"]
```

Followup:
Given a string. You have to split the string based on space but ignore those spaces that are preceded by backslash. Eg:

```
str = "I love\ India"
output: ["I", "love India"]
```

Expectation here was to ask a lot of clarifying questions from the interviewer (as all the cases are not covered in the example) and then start the implementation.

Design rope

Design rope data structure 😨

Number plate

The number on the number plate of a vehicle has alphanumeric characters. The number is a string of 6 characters of which first 2 characters are alphabets while last 4 are digits. The first number generated is *AA0000* while the last number generated is *ZZ9999*. Find the kth number generated.

Matrix block

Given a 2D grid with n rows and m columns. Some cells are blocked which you cannot pass through. From each cell you can go either up, down, left or right. Find the shortest path from (0, 0) to (n-1, m-1).

Followup:

Now assume that there exists no path from (0, 0) to (n-1, m-1) (All paths are blocked). Find the minimum number of cells that you need to unblock such that there exists a path from (0, 0) to (n-1, m-1). Can you solve it in O(n+m)

Google telephonic interview questions

Least weighted common ancestors path

Given a tree where each edge has some weights. Now given 2 nodes n1 and n2, Find out the minimum weight in the common path from root to n1 and n2.

Follow-up:
How will you efficiently solve it if the number of such queries are huge?

ans :-> pre-process and binary lifting :)

K stack pops

K stack pops

Lawn mower

(one of the hardest questions. Too big hence please remind to update)

Randomizer

Given a class like the following:

```
class Randomizer {

      int n, k;
      Randomizer(int n, int k) {
            this.n = n;
            this.k = k;
      }
```

```
      int getRandom() {


      }
}
```

Implement the getRandom method. getRandom returns a random number in the range [1,n] but a number once generated should not be generated again in next k turns. For example:

n = 5
k = 3

getRandom() returns 2 (assume)
getRandom() return x
getRandom() returns y
getRandom() returns z
getRandom() returns w

now as per the constraint x, y and z cannot be 2. But w can be 2. Similarly y, z and w cannot be x.

Topological ordering

Given task dependencies (x, y) denoting that to complete y one must first complete x. Find the least amount of time needed to finish all the tasks given that the task that can be done in parallel can be processed together and it takes 1 unit of time to process a task.

Facebook interview questions

Palindromic string

Given a string. Tell if it is possible to make this string a palindrome by removing just one character.

Nodes in a range in a binary tree

Given the root of the binary search tree and a range [l, r]. Return sum of elements that lie within this range.

Pattern searching

Given a string and a pattern. Tell if the string follows the pattern. For example:

String = "Facebook", Pattern = "F6k"
Output: true

String = "Rajat", Pattern = "R3"
Output: false

Minimum removals to make balanced string

Link

Airbnb interview questions

In memory file system

question
Followup: Trigger a callback on successful get operation

Palindromic pairs

question

Banking service

A banking application has following APIs:

addMoney(accountId, amount, time)   :-> adds amount to account id

deductMoney(accountId, amount, time) :-> deduct amount from account id

getBalance(accountId) :-> fetches latest balance in account id

getBalanceDiff(accountId, time1, time2) :-> finds the difference between the balances between time2 and time1. Note that at time1 and time2 balance may not have been updated. The task is to find the difference between the balance at time2 and time1.

The first 3 APIs should work in constant time while the fourth one should be in logarithmic time.

Design queue

Design a queue using an array of size no more than 5.

This was the weirdest question out there. I had to ask following questions to complete this one:

1. If an array has size 5 then are we not allowed to add more than 5 elements? The answer was no. The queue can have infinite capacity.
2. Can we use a 2D array and add a new array when the size of the current crosses 5? The answer was no. Since a 2D array is also an array and its size can cross 5.
3. The final solution was to use a linked list with each node having a 1D array. And when the size crosses 5 we add a new node. 🤷

Pour water

Please note that this question is different from trapping rain water.