**Goldman interview**

First interview:

**Thomas Rosal** & **Jin Meng**

Thomas Rosal is more on the corporate side (in the investment banking division).

Jin Meng is in the Asset Management Division. Example of project: fixed income portfolio management. They get guidelines from clients and transform into code constrains, they also take market constrains and Goldman constraints into account.

Rebalance the portfolio every …, they need to make sure they have enough computation capacity. Then, they also need to deal with execution.

Why Goldman? emphasize on collaboration, you need to know your team members, where you can get help.

Thomas gave the example of a time during his internship when he was struggling on a project at 9PM and an associate stayed with him to help him.

Math question: you have a dice; you throw it twice. After your first throw, you can either keep the number or throw it again but you will lose your first result. Goal: getting the highest number. What is the expected value?

My answer: your expected value after the first throw is 3.5=. The strategy is to continue only if your first throw is below your expected value (which is 3.5). Therefore, for 1, 2, 3, you throw it again and you have an expected value of 3.5 for them too. Your final expected value is =4.25.

A bit harder: you now throw it 3 times. What is your strategy?

My answer: Given the precedent value, 4.25. I throw the first dice, if I get less than 4.25 (1, 2 3 and 4), I throw it again and I throw a third time if I get less than 3.5 (the expected value of the last throw, i.e. for 1, 2 and 3). No need to do the calculation but the answer is.

A bit harder: you now throw it n times, n large.

My answer: if n goes to infinity, my expected value would be 6, I would only stop once I get 6 and I will eventually get it if n goes to infinity.

My comment for n fixed: I would do it recursively, for n I need n-1 and so on (as I did with 3).

Behavioural question: about projects I did, my internships …

Most interesting course that I had: Non-Linear Option Pricing, I talked a bit about it.

Second interview:

Kaveen Herath Bendara: private sector

About the culture: same as before.

Behavioural part: same kind as interview 1. Also, if I had a or several projects with a 1-month deadline, how do I organize? to do list …

Math question:

You have 1000 coins and one is unfair (it has 2 heads). You throw a coin 10 times and you get 10 heads, what is the probability that this coin is unfair?

My answer:

*Notation*: A: unfair; B: 10 head in a row

P(A|B)=P(A and B)/P(B)

P(B)=P(B|A)P(A) + P(B| no A)P(no A) 1/1000 + 1/1000

P(B|A) = 1

P(A)=1/1000

P(B| no A)=1/(2^10)=1/1024

P(no A)=999/1000

P(A and B)=P(B|A)P(A) = 1/1000

Answer approximative: 0.5. exact =

Coding question:

You have a sparse vector, how to store it? How to do a dot product of two sparse vector once stored, how to add 2 sparse vector.

If you have to do the reverse, create the original list from the new data structure, is it possible? no, you need to know the 0 at the end. To solve that, you can add a parameter, the size of the initial vector.

#

l=[0,0,0,0,0,0,0,1,0,0,0,12,2,6,0,0,0,0,0]

def sparseVector(nums: List[int]):

d={}

for i in range(len(nums)):

if nums[i]!=0:

d[i]=nums[i]

return d

def dotProduct(sparseVector1, sparseVector2):

res = 0

for x in sparseVector1:

if x in sparseVector2:

res+= sparseVector1[x]\*sparseVector2[x]

return res

from collections import Counter

def add(sparseVector1, sparseVector2) :

res=0

d = Counter()

for x in sparseVector1:

d[x] += sparseVector1[x]

for y in sparseVector1:

d[y]+= sparseVector2[y]

return res