JP Morgan

2020 Mar 30 : Kennant (system design in CASH) and Roy (C++ questions)
2020 Apr 03 : Daniel (market making game) and Simon (hyperthetical crisis management)
2020 Apr 16 : Jitendra (option pricing) and Desmond (nothing)

Question 1
Given a fair dice, you can roll it 3 times atmost, you can choose to stop right after each roll, the payoff will be the realized value of the last roll. What is fair price to enter this game?

Answer
Build 3 layer-network with 6 nodes in each layer, write down the payoff at each node if it is exercised, then do backward propagation with updating equation :

```
value[time, state] = max(payoff_exercised_now, E[value[time+1, state]])
```

Question 2
Given a fair coin, you can flip it 1000 times atmost, you can choose to stop right after each flip, the payoff will be the total number of heads you got divided by total number of flips you did. What is the fair price to enter this game?

Answer
Normally, coin flip sequence is Markov chain, which can be formulated as a binomial tree. Depending on the payoff :

- if payoff is path-dependent, then it is a non-combining binary tree (like finding num flips to get 1st head)
- if payoff is not path-dependent, then it is a combining binary tree (like finding num of heads in total)
- of course, combining tree saves space and time
- in short, this question is the same as the previous one, just plug in a new Markov chain and new payoff

For example, consider sequence HHHHTTTT and TTTTHHHH :

	Num of flip to first head	Num of head in total
HHHHTTTT	1	4
TTTTHHHH	5	4
	This is path dependent.	This is not.

This coin game can be regarded as an American option with underlying dynamics and derivative payoff defined as :

$$
\begin{aligned}
S_0 &= 0 \\
S_t &= \left[\begin{array}{ll} S_{t-1}+1 & p = 0.5 \\ S_{t-1} & q = 0.5 \end{array}\right. \\
payoff\,(S_t) &= S_t / t
\end{aligned}
$$

This American option can be priced numerically using combining binomial tree :

```
def coin_game_price(maturity) :
    price = []
    for n in range(maturity+1) : price.append(n/maturity)
    for t in range(maturity-1,0,-1) :
        price1 = []
        for n in range(len(price)-1) : price1.append(max(n/t, (price[n]+price[n+1])/2.0))
        price = list(price1)

    if len(price) != 2 : print('ERROR')
    return (price[0]+price[1])/2

for t in [10,100,1000,2000,5000,10000,20000] : print('price for maturity ', t, '=', coin_game_price(t))
```

Crisis management questions

What happen when system crash?

1. alternative to get mktdata (reconnection)
2. alternative to get position (drop copy from HKEX)
3. alternative to place order (3rd party broker supply system)
4. inform the team
5. enough test to ensure crash prob near zero
6. enough contingency plan, such as :

- reboot system and recover original state
- more than one path to get mktdata
- more than one path to get position
- more than one path to place order

Answer from ex-EMM team member

1. check if there are problems for other traders
2. cancel outstanding orders
3. reboot system in parallel

What happen when there is price discrepancy between two channels of market datafeed?

1. if one channel is real time quote while another is prediction, then arbitrage
2. if both channels are real time quote, then :

- if price difference is large, check against 3rd party
- if price difference is small, find lag, trust the fast one
- if price difference is small, but not a lag relationship, then ...

→ there must be bugs in one of them, check against 3rd 4th ... party and hedge position, start debugging

Answer from ex-EMM team member

1. trust bloomberg as it is fairly accurate