一、

只有做Bonus2。

二、

(1)Black-Scholes : **Option\_BlackScholes.py**

def bs(S0, K, T, r, q, sigma, call\_put):...

# main

S0 = 50

K = 50

r = 0.1

q = 0.05

sigma = 0.4

T = 0.5

testCall = bs(S0, K, T, r, q, sigma, "call")

testPut = bs(S0, K, T, r, q, sigma, "put")

print('========================================')

print(testCall)

print(testPut)

直接調整 **#main** 下面的變數，呼叫函式存入變數，再印出即可(所有參數以及函式都已預先輸入好，直接執行即可以看到精美的結果)。

(2)Monte-Carlo Simulation : **Option\_MonteCarlo2.py**

def monte\_carlo\_European(S0, K, T, r, q, sigma, call\_put, sims, rep):..

# main

S0 = 50

K = 50

r = 0.1

q = 0.05

sigma = 0.4

T = 0.5

sims = 10000

rep = 20

monte\_carlo\_European(S0, K, T, r, q, sigma, "call", sims, rep)

monte\_carlo\_European(S0, K, T, r, q, sigma, "put", sims, rep)

直接調整 **#main** 下面的變數，再呼叫函式即可(所有參數都以及函式已預先輸入好，直接執行即可以看到精美的結果)。

(3)CRR　Binomial Tree : **Option\_BinomialTree.py**

**\*Bonus2已包含在 Option\_BinomialTree.py 中**

from math import log, sqrt, exp

# from scipy.stats import binom

def binomial\_prob(n, j, p):...

def binomial\_European(S0, K, T, r, q, sigma, layers, call\_put):...

def binomial\_American(S0, K, T, r, q, sigma, layers, call\_put):...

def combinatorial\_European(S0, K, T, r, q, sigma, layers, call\_put):...

# main

S0 = 50

K = 50

r = 0.1

q = 0.05

sigma = 0.4

T = 0.5

# n = 100 (layers = 100)

layers = 100

# European Call

...

直接調整 **#main** 下面的變數，再呼叫函式即可(所有測資(不同的n)、參數以及函式都已預先輸入好，直接執行即可以看到相當精美的結果)。

**直接執行，輸出看起來是這樣 :**

**============================================================**

**n = 100**

**(CRR Binomial Tree) Price of European call : 6.026046**

**(CRR Binomial Tree) Price of American call : 6.026232**

**(CRR Binomial Tree) Price of European put : 4.822022**

**(CRR Binomial Tree) Price of American put : 4.97949**

**============================================================**

**n = 500**

**...**

**關於Bonus2 :**

直接呼叫 **Option\_BinomialTree.py 中**

**combinatorial\_European(S0, K, T, r, q, sigma, layers, call\_put)**

即可 (已經幫老師/助教呼叫好，直接執行即可)。

**最後，由CRR Binomial Tree與Monte-Carlo Simulation所計算出來的價格都已四捨五入至第六位**