IE523 B,OB,ONL: Financial Computing
Fall, 2022
Programming Assignment 9: Pricing European- and American-Options Using the Trinomial Model with Memoization
Due Date: 18 November 2022

In the previous programming assignment you used Dynamic Programming to compute the price of an American-Option to overcome the inefficiency of Recursive methods that do the same. You were able to comfortably run large number of Divisions, to get at an accurate estimate of the price.

In this assignment, you are going to make the Recursive methods efficient by using *Memoization*. You have done this already in the Programming Assignment that computed the price-of-admission to a stylized Card-Game. You are going to do the same – only this time, it is for Recursive procedures that compute the price of Options.

Write a C++ program that computes the price of an

1. European Option, and
2. American Option,

which runs on a command-line, and takes the following variables as input (maintain this order; otherwise, you will loose points!)

1. Expiration,
2. #Stages,
3. Risk-free rate (Continuous compounding) ,
4. Volatility (expressed as a fraction of initial price of underlying),
5. Initial Price of underlying, and
6. Strike Price.

using the Trinomial Model. It is imperative that your code uses memoization, so that we can run large number (≈ 1000) of stages.

The European Option Price has be compared with the Black-Scholes Formula (cf. figure 1). For the American Option Price, I am looking for something like what is shown in figure 2. You can ignore the *put-call parity* verification, if you wish.

```
Debug — bash — 98×27
mobile-96-74:Debug sreenivas$ time ./European\ via\ Memoized\ Trinomial 0.5 5000 0.08 0.3 60 50
(Memoized) Recursive Trinomial European Option Pricing
Expiration Time (Years) = 0.5
Number of Divisions = 5000
Risk Free Interest Rate = 0.08
Volatility (%age of stock value) = 30
Initial Stock Price = 60
Strike Price = 50
--------------------------------------
Up Factor = 1.00425
Uptick Probability = 0.250413
--------------------------------------
Trinomial Price of an European Call Option = 12.8225
Call Price according to Black-Scholes = 12.8226
--------------------------------------
Trinomial Price of an European Put Option = 0.862011
Put Price according to Black-Scholes = 0.862071
--------------------------------------
Verifying Put-Call Parity: S+P-C = Kexp(-r*T)
60 + 0.862011 - 12.8225 = 50exp(-0.08 * 0.5)
48.0395 = 48.0395
--------------------------------------

real    0m3.364s
user    0m3.115s
sys     0m0.203s
mobile-96-74:Debug sreenivas$
```

Figure 1: Running-time illustration for pricing an European Option using a 5000 stage (Memoized) Trinomial Model.

```
Debug — bash — 104×27
mobile-96-74:Debug sreenivas$ time ./American\ Option\ via\ Memoized\ Trinomial 0.5 5000 0.08 0.3 60 50
(Memoized) Recursive Trinomial American Option Pricing
Expiration Time (Years) = 0.5
Number of Divisions = 5000
Risk Free Interest Rate = 0.08
Volatility (%age of stock value) = 30
Initial Stock Price = 60
Strike Price = 50
--------------------------------------
Up Factor = 1.00425
Uptick Probability = 0.250413
Notick Probability = 0.5
Downtick Probability = 0.249588
--------------------------------------
Trinomial Price of an American Call Option = 12.8225
Trinomial Price of an American Put Option = 0.896696
--------------------------------------
Let us verify the Put-Call Parity: S+P-C = Kexp(-r*T) for American Options
60 + 0.896696 - 12.8225 = 50exp(-0.08 * 0.5)
48.0742 ?=? 48.0395
Looks like Put-Call Parity does NOT hold
--------------------------------------

real    0m7.595s
user    0m7.363s
sys     0m0.207s
mobile-96-74:Debug sreenivas$
```

Figure 2: Running-time illustration for pricing an American Option using a 5000 stage (Memoized) Trinomial Model.