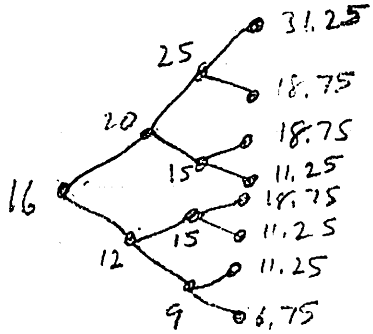
Rohit Penumarti
Section: M16-6:50
4/27/20

176 HW #1

2.



$r = 0.05$
$u = 1.25$
$d = 0.75$

$V_3 = \left(\frac{1}{3}Y_3 - S_3\right)^+$

$\hat{p} = \frac{1+r-d}{u-d} = \frac{1.05-0.75}{1.25-0.75} = 0.6$

$\hat{q} = 1-\hat{p} = 0.4$

$V_3(HHH) = \left(\frac{20+25+31.25}{3} - 31.25\right)^+ = (25.4 - 31.25)^+ = 0$

$V_3(HHT) = \left(\frac{20+25+18.75}{3} - 18.75\right)^+ = (21.25 - 18.75)^+ = 2.5$

$V_3(HTH) = \left(\frac{20+15+18.75}{3} - 18.75\right)^+ = (17.9 - 18.75)^+ = 0$

$V_3(HTT) = \left(\frac{20+15+11.25}{3} - 11.25\right)^+ = (15.41 - 11.25)^+ = 4.17$

$V_3(THH) = \left(\frac{12+15+18.75}{3} - 18.75\right)^+ = (15.25 - 18.75)^+ = 0$

$V_3(THT) = \left(\frac{12+15+11.25}{3} - 11.25\right)^+ = (12.75 - 11.25)^+ = 1.5$

$V_3(TTH) = \left(\frac{12+9+11.25}{3} - 11.25\right)^+ = (10.75 - 11.25)^+ = 0$

$V_3(TTT) = \left(\frac{12+9+6.75}{3} - 6.75\right)^+ = (9.25 - 6.75)^+ = 2.5$

$$V_0 = \frac{\tilde{E}(V_3)}{(1+r)^3} = \frac{(0.6)^2(0.4)(2.5) + 0.6(0.4^2)(4.17) + 0.4^2(0.6)(1.5) + 0.4^3(2.5)}{(1+0.05)^3}$$

$$\boxed{V_0 = 0.919}$$

The replicating portfolio at each time step $t=0,1,2$ for $\omega = THT$ is given by

$$\Delta_n(\omega_1,\dots,\omega_n) = \frac{V_{n+1}(\omega_1,\dots,\omega_n H) - V_{n+1}(\omega_1,\dots,\omega_n T)}{S_{n+1}(\omega_1,\dots,\omega_n H) - S_{n+1}(\omega_1,\dots,\omega_n T)} = \frac{V_{n+1}(H) - V_{n+1}(T)}{(u-d)S_n}$$

$$\Delta_2(TH) = \frac{V_3(THH) - V_3(THT)}{(u-d)S_2} \quad \frac{0-1.5}{(1.25-0.75)15} = \boxed{-0.2}$$

$V_2(TH) = \frac{1}{1+0.05}[0.6(0) + 0.4(1.5)] = 0.5714$

$V_2(TT) = \frac{1}{1.05}[0.6(0) + 0.4(2.5)] = 0.9524$

$$\Delta_1(T) = \frac{V_2(TH) - V_2(TT)}{(u-d)S_1} = \frac{0.5714 - 0.9524}{0.5(12)} = \boxed{-0.0635}$$

$V_1(T) = \frac{1}{1.05}[0.6(0.5714) + 0.4(0.9524)] = 0.689$

$V_1(H) = \left(\frac{1}{1.05}\right)^2[0.6^2(0) + 0.24(2.5) + 0.24(0) + 0.4^2(4.17)] = 1.1489$

$$\Delta_0 = \frac{V_1(H) - V_1(T)}{(u-d)S_0} = \frac{1.1489 - 0.689}{0.5(16)} = \boxed{0.0574}$$

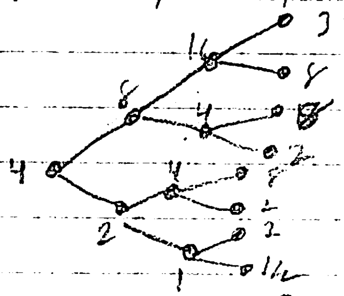
Where each Δ value denotes the number of stocks to buy (or sell). Now, for the amount to invest, refer to the below values:

$$V_2(TH) - \Delta_2(TH) \cdot S_2(TH) = 0.5714 + 0.2(15) = \boxed{3.57}$$

$$V_1(T) - \Delta_1(T) \cdot S_1(T) = 0.689 + 0.0635(12) = \boxed{1.45}$$

$$V_0 - \Delta_0 S_0 = 0.919 - 0.0574(16) = \boxed{0}$$

3. $S_0=4,\ u=2,\ d=1/2,\ r=1/4,\ \tilde{p}=\tilde{q}=1/2,\ K=4$



$V_3(s,y)=\left(\frac{1}{4}Y_3-K\right)^+$ where $Y_n=\sum_{k=0}^{n} S_k$

i) $V_n(s,y)=\frac{1}{1+r}\left[\tilde{p}V_{n+1}(us,y+us)+\tilde{q}V_{n+1}(ds,y+ds)\right]$

$=\frac{1}{1+1/4}\left[\frac{1}{2}V_{n+1}(2s,y+2s)+\frac{1}{2}V_{n+1}(\tfrac{1}{2}s,y+\tfrac{1}{2}s)\right]$

$=\frac{4}{5}\left[\frac{1}{2}V_{n+1}(2s,y+2s)+\frac{1}{2}V_{n+1}(\tfrac{1}{2}s,y+\tfrac{1}{2}s)\right]$

$$\boxed{V_n(s,y)=\frac{2}{5}\left[V_{n+1}(2s,y+2s)+V_{n+1}(\tfrac{1}{2}s,y+\tfrac{1}{2}s)\right]}$$

ii)

Path	Y_3	V_3
HHH	60	11
HHT	36	5
HTH	24	2
HTT	18	0.5
THH	18	0.5
THT	12	0
TTH	9	0
TTT	7.5	0

Path	V_2
HH	6.4
HT	1
TH	0.2
TT	0

Path	V_1
H	2.96
T	0.08

$V_0(4,4)=\frac{2}{5}(2.96+0.08)$

$$\boxed{V_0(4,4)=1.216}$$

iii) $\delta_n(s,y)=\dfrac{V_{n+1}(us,y+us)-V_{n+1}(ds,y+ds)}{(u-d)s}$

$$\boxed{\delta_n(s,y)=\frac{V_{n+1}(2s,y+2s)-V_{n+1}(\frac{1}{2}s,y+\frac{1}{2}s)}{(3/2)s}}$$

Homework #1 Code

April 27, 2020

```
In [1]: import numpy as np
        from itertools import product
        import pandas as pd
        import math

In [97]: # Problem 4
         S0 = 10
         u = 1.05
         d = 0.95
         r = 0.01
         N = 8
         K = 11
         q = (1+r-d)/(u-d)

         w = np.array(list(product((0.95,1.05), repeat=8)))
         SN = S0*np.cumprod(w, axis=1)
         SN = np.insert(SN, 0, S0, axis=1)

         B = np.array([11, 11.5, 12, 12.5, 13])
         payoff = np.zeros((np.size(SN, 0), np.size(B)))
         non_zero_count = np.zeros(np.size(B))

         numH = (w == 1.05).sum(axis=1)
         numT = 8-numH
         risk_neutral_prob = np.zeros((np.size(SN, 0), np.size(B)))
         risk_neutral_prob_B = np.zeros(np.size(B))
         price = np.zeros(np.size(B))

         for i in range(np.size(B)):
             for j in range(np.size(SN, 0)):
                 if(np.amax(SN[j]) > B[i]):
                     risk_neutral_prob[j,i] += q**numH[j]*(1-q)**numT[j]
                     if(SN[j,8] > K):
                         payoff[j,i] = SN[j,8]-K
                         price[i] += risk_neutral_prob[j,i]*payoff[j,i]
                         if(payoff[j,i] != 0):
                             non_zero_count[i] += 1
             risk_neutral_prob_B[i] = np.sum(risk_neutral_prob[:,i])
```

```
    price = price/(1+r)**8
    results = {'B Value':B, 'Non-zero Count':non_zero_count, 'Risk-Neutral Prob':risk_neu

    df = pd.DataFrame(results)
    dfStyler = df.style.set_properties(**{'text-align': 'center'})
    dfStyler.set_table_styles([dict(selector='th', props=[('text-align', 'center')])])
    dfStyler
```

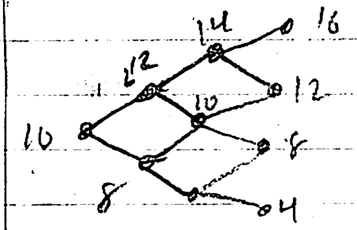
Out[97]: <pandas.io.formats.style.Styler at 0x25342b0f6d8>

```
In [44]: # Problem 6 Extra Credit
         S0 = 1.05
         u = 1.01
         d = 0.99
         r = 0
         q = (1+r-d)/(u-d)

         def sim_N_period_binom(N):
             w = np.array(list(product((d,u), repeat=N)))
             SN = S0*np.cumprod(w, axis=1)
             SN = np.insert(SN, 0, S0, axis=1)
             return SN

         def hit_box_option_payoff(t, S, N):
             t1 = math.ceil(t[0])
             t2 = int(t[1])

             SN = sim_N_period_binom(N)
             SN_box = SN[:, t1:t2+1]

             box_check = np.where(np.logical_and(SN_box>=S[0], SN_box<=S[1]), 1, 0)
             payoff = np.any(box_check, axis=1)

             price = q**N*np.sum(payoff)/(1+r)**N
             return price

         def miss_box_option_payoff(t, S, N):
             t1 = math.ceil(t[0])
             t2 = int(t[1])

             SN = sim_N_period_binom(N)
             SN_box = SN[:, t1:t2+1]

             box_check = np.where(np.logical_or(SN_box<=S[0], SN_box>=S[1]), 0, 1)
             payoff = 1-np.any(box_check, axis=1)

             price = q**N*np.sum(payoff)/(1+r)**N
```

```python
    return price

print("The price of the Hit Box Option is: " + str(hit_box_option_payoff([1.9, 4.1],
print("The price of the Miss Box Option is: " + str(miss_box_option_payoff([1.9, 4.1]
```

```
The price of the Hit Box Option is: 0.75
The price of the Miss Box Option is: 0.375
```

5. $r = 0.05$, $K = 11$



$$V_3 = (S_T - K)^+ + 1.50\mathbb{1}_{\{\max S_n < K\}}$$

$V_3(HHH) = (16-11)^+ + 1.50\mathbb{1}_{\{16<11\}} = 5$

$V_3(HHT) = (12-11)^+ + 1.50\mathbb{1}_{\{14<11\}} = 1$

$V_3(HTH) = (12-11)^+ + 1.50\mathbb{1}_{\{12<11\}} = 1$

$V_3(HTT) = (8-11)^+ + 1.50\mathbb{1}_{\{12<11\}} = 0$

$V_3(THH) = (12-11)^+ + 1.50\mathbb{1}_{\{12<11\}} = 1$

$V_3(THT) = (8-11)^+ + 1.50\mathbb{1}_{\{10<11\}} = 1.50$

$V_3(TTH) = (8-11)^+ + 1.50\mathbb{1}_{\{10<11\}} = 1.50$

$V_3(TTT) = (4-11)^+ + 1.50\mathbb{1}_{\{10<11\}} = 1.50$

Path	u	d	$\tilde{p}$	$\tilde{q}$
HHH	8/7	N/A	0.675	0.325
HHT	N/A	6/7		
THH HTH	1.2	N/A	0.625	0.375
THT HTT	N/A	0.8		
TTH	4/3	N/A	0.575	0.425
TTT	N/A	2/3		
HH	7/6	N/A	0.65	0.35
HT	N/A	5/6		
TH	5/4	N/A	0.6	0.4
TT	N/A	3/4		
H	1.2	N/A	0.625	0.375
T	N/A	0.8		

$V_2(HH) = \frac{1}{1.05}[0.675(5) + 0.325(1)]$
$= 3.524$

$V_2(HT) = \frac{1}{1.05}[0.625(1) + 0.375(0)]$
$= 0.5952$

$V_2(TH) = \frac{1}{1.05}[0.625(1) + 0.375(1.50)]$
$= 1.131$

$V_2(TT) = \frac{1}{1.05}[0.575(1.50) + 0.425(1.50)] = 1.4286$

$V_1(H) = \frac{1}{1.05}[0.65(3.524) + 0.35(0.5952)]$
$= 2.3798$

$V_1(T) = \frac{1}{1.05}[0.6(1.131) + 0.4(1.4286)] = 1.1905$

$V_0 = \frac{1}{1.05}[0.625(2.3798) + 0.375(1.1905)]$

$$\boxed{V_0 = 1.84}$$

The payoff formula is simply the vanilla call option payoff added with the compensatory coupon payment of \$1.50 multiplied with an indicator function. The condition in the indicator function sees if the maximum along a path is less than K. This works because if the max along a path is less than K, then it will never reach the strike price whereas if the max along a path is greater than K, then you know for certain that it reached K.