1(a) $A = \begin{pmatrix} 1.5 & 4 \\ 1 & 1 \end{pmatrix}$ has rank 2 $\Rightarrow$ The market is always dynamically complete

(b)(c) $R_u = 2$ $R_d = 0.5$ $R_f = 1.5$

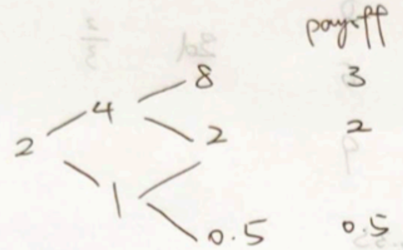
$$q_u = \frac{R_f - R_d}{R_u - R_d} = \frac{2}{3} \qquad q_d = \frac{R_u - R_f}{R_u - R_d} = \frac{1}{3}$$

$$\psi_u = \frac{q_u}{R_f} = \frac{4}{9} \qquad \psi_d = \frac{q_d}{R_f} = \frac{2}{9}$$

(d) $P_{nk} = \psi_{nk} R_f^n = C_n^{k-1} \left(\frac{4}{9}\right)^{n+1-k} \left(\frac{2}{9}\right)^{k-1} \left(\frac{3}{2}\right)^n = C_n^{k-1} \left(\frac{2}{3}\right)^{n+1-k} \left(\frac{1}{3}\right)^{k-1}$

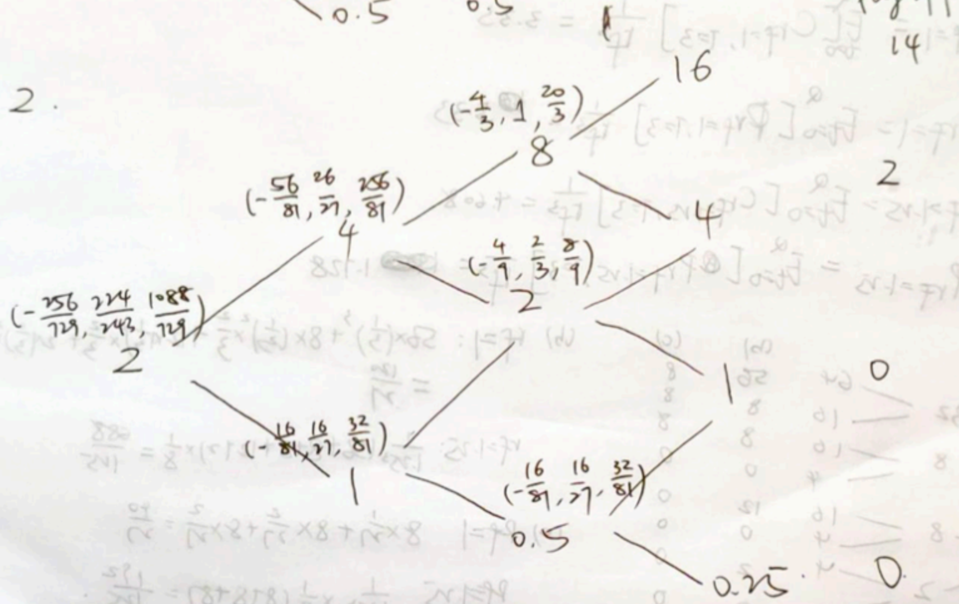
(e) $\bar{R} = \frac{1}{3} \times 2 + \frac{2}{3} \times 1 = 1$ risk premium $= 1 - 1.5 = -50\%$

(f)



$$\text{price} = \frac{\frac{4}{9} \times 3 + \frac{2}{9} \times 2 + \frac{1}{9} \times \frac{1}{2}}{1.5^2} = \frac{82}{81}$$

2.



Three elements in brackets are (B, S, v) respectively

(b) $\frac{1088}{729}$ (c) They should be equal since payoffs are perfectly same and there is no arbitrage

3 (a) $E[P/L] = \frac{1}{27}(P-14) + \frac{6}{27}(P-2) + \frac{20}{27}P = P - \frac{26}{27} = V + \frac{1}{27} = \frac{1115}{1729}$

$\operatorname{var}(P/L) = \frac{1}{27}\left[(P-14) - (P - \frac{26}{27})\right]^2 + \frac{6}{27}\left[(P-2) - (P - \frac{26}{27})\right]^2 + \frac{20}{27}\left[P - (P - \frac{26}{27})\right]^2$

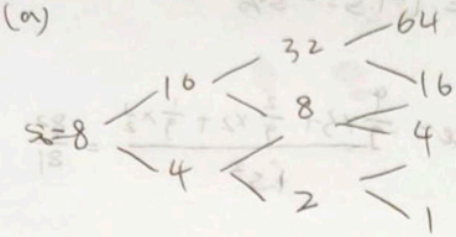
$= 6.56$

(b) I can sell the option and long the replicating portfolio

I will have +1 profit today with 0 volatility at any time.

My P/L will not depend on market price at intermediate times.

4. (a)



	call	put
64	54	0
16	6	0
4	0	6
1	0	9

	$R_f=1$	$R_f=1.25$
q_u	$\frac{1}{3}$	$\frac{1}{2}$
q_d	$\frac{2}{3}$	$\frac{1}{2}$

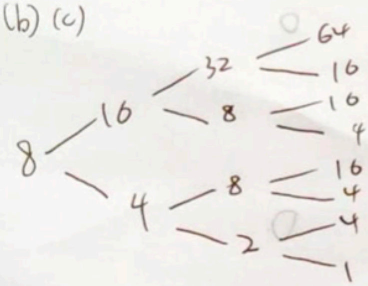
$C_{rf=1} = E^Q_{t=0}[C_{rf=1, T=3}] \frac{1}{r_f^3} = 3.33$

$P_{rf=1} = E^Q_{t=0}[P_{rf=1, T=3}] \frac{1}{r_f^3} = 5.33$

$C_{rf=1.25} = E^Q_{t=0}[C_{rf=1.25, T=3}] \frac{1}{r_f^3} = 4.608$

$P_{rf=1.25} = E^Q_{t=0}[P_{rf=1.25, T=3}] \frac{1}{r_f^3} = 1.728$

(b) (c)



	(b)	(c)
64	56	8
16	8	8
16	8	8
4	0	0
16	12	0
4	0	0
4	2	0
1	0	0

(b) $r_f=1$: $56\times(\frac{1}{3})^3 + 8\times(\frac{1}{3})^2\times\frac{2}{3} + 12\times(\frac{1}{3})^2\times\frac{2}{3} + 2\times(\frac{2}{3})^2\times\frac{1}{3}$
$= \frac{120}{27}$

$r_f=1.25$: $\frac{1}{1.25^3}(56+8+8+12+2)\times\frac{1}{8} = \frac{688}{125}$

(c) $R_f=1$ $8\times\frac{1}{27} + 8\times\frac{2}{27} + 8\times\frac{2}{27} = \frac{40}{27}$

$R_f=1.25$ $\frac{1}{1.25^3}\times\frac{1}{8}(8+8+8) = \frac{192}{125}$.

```
In [65]: def BTvalues(S, K, T, Ru, Rd, Rf):
    ...:     def comb(n,m):
    ...:         return math.factorial(n)/(math.factorial(m)*math.factorial(n-m))
    ...:
    ...:     qu = (Rf - Rd) / (Ru - Rd)
    ...:     qd = (Ru - Rf) / (Ru - Rd)
    ...:     CT = list()
    ...:     PT = list()
    ...:     qf = list()
    ...:     for i in range(0,T+1):
    ...:         ST = (Ru**(T-i))*(Rd**i)*S
    ...:         qf.append((qu**(T-i))*(qd**i)*comb(T,i))
    ...:         if ST > K:
    ...:             CT.append(ST - K)
    ...:             PT.append(0)
    ...:         else:
    ...:             CT.append(0)
    ...:             PT.append(K - ST)
    ...:     CT = np.array(CT)
    ...:     qf = np.array(qf)
    ...:     c = CT@qf/(Rf**T)
    ...:     p = PT@qf/(Rf**T)
    ...:     return c,p
    ...: def delta(S, K, T, Ru, Rd, Rf):
    ...:     c, p= BTvalues(S, K, T, Ru, Rd, Rf)
    ...:     Su = S*Ru
    ...:     Sd = S*Rd
    ...:     vu,pu = BTvalues(Su,K,T-1,Ru,Rd,Rf)
    ...:     vd,pd = BTvalues(Sd,K,T-1,Ru,Rd,Rf)
    ...:     delta_c = (vu - vd)/(Su-Sd)
    ...:     delta_p = (pu - pd)/(Su-Sd)
    ...:     return delta_c, delta_p
    ...:
    ...:
    ...: c,p = BTvalues(100,105,3,2,0.5,1.25)
    ...: delta_c,delta_p = delta(100,105,3,2,0.5,1.25)

In [66]: c
Out[66]: 62.72

In [67]: p
Out[67]: 16.48

In [68]: delta_c
Out[68]: 0.8426666666666666

In [69]: delta_p
Out[69]: -0.15733333333333333
```