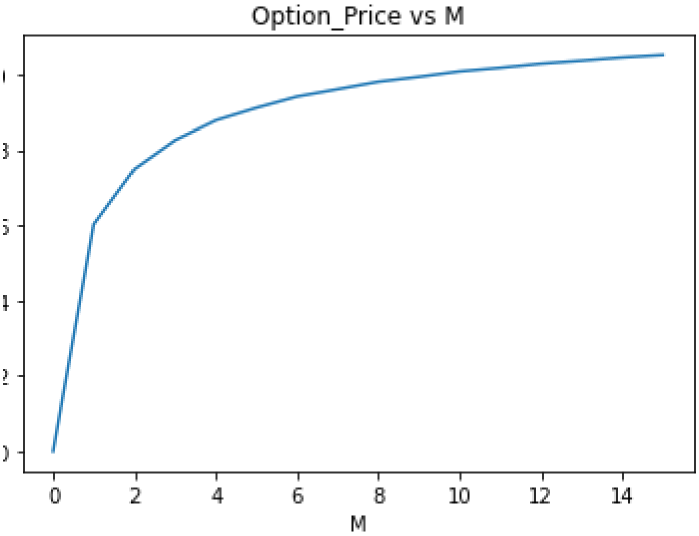
```
import numpy as np
import matplotlib.pyplot as plt


S_0 = 100
T = 1
r = 0.08
sigma = 0.2
M_values = [5, 10, 15]
option_price = [ ]

def lookback_matrix(M, u, d):
    Payoff = [[[S_0, S_0]]]
    for a in range(M):
        S = [ ]
        for b in range(len(Payoff[a])):
            Q = Payoff[a][b][0]*u
            Q_max = max(Payoff[a][b][1], Q)
            S.append([Q, Q_max])
            Q = Payoff[a][b][0]*d
            Q_max = max(Payoff[a][b][1], Q)
            S.append([Q, Q_max])
        Payoff.append(S)
    return Payoff

def lookback_option(M, u, d, p):
    Payoff = [[[S_0, S_0]]]
    for a in range(M):
        S = [ ]
        for b in range(len(Payoff[a])):
            Q = Payoff[a][b][0]*u*p
            Q_max = p*max(Payoff[a][b][1], Q/p)
            S.append([Q, Q_max])
            Q = Payoff[a][b][0]*d*(1-p)
            Q_max = (1-p)*max(Payoff[a][b][1], Q/(1-p))
            S.append([Q, Q_max])
        Payoff.append(S)
    sol = 0
    for p1 in Payoff[len(Payoff)-1]:
        sol += (p1[1]-p1[0])
    return (sol)*np.exp(-r*T)

for M in M_values:
    dt = T/M
    u = np.exp(sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
    d = np.exp(-sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
    p = (np.exp(r*dt)-d)/(u-d)
    if(p<0 or p>1):
        print('For M = ', M, 'the no arbitrage principle is violated.')
        continue
    Value = lookback_option(M, u, d, p)
    option_price.append(Value)
    print('For M = ', M, 'the lookback option value is :', Value)
```

```
    print('For M = ', M, 'the lookback option value is : ', value)

M_values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
option_price = [0]

for M in M_values[1:]:
    dt = T/M
    u = np.exp(sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
    d = np.exp(-sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
    p = (np.exp(r*dt)-d)/(u-d)
    if(p<0 or p>1):
        print('For M = ', M, 'the no arbitrage principle is violated.')
        continue
    Value = lookback_option(M, u, d, p)
    option_price.append(Value)

plt.plot(M_values, option_price)
plt.xlabel('M')
plt.ylabel('Lookback Option Price')
plt.title('Option_Price vs M')
plt.show()


M = 5
dt = T/M
u = np.exp(sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
d = np.exp(-sigma*np.sqrt(dt) + (r-0.5*sigma*sigma)*dt)
p = (np.exp(r*dt)-d)/(u-d)
if(p<0 or p>1):
    print('For M = ', M, 'the no arbitrage principle is violated')
else:
    mat = lookback_matrix(M, u, d)
    price_list = []
    for Payoff in mat[len(mat)-1]:
        price_list.append(Payoff[1]-Payoff[0])
    for a in range(6):
        print('Time point t = ', (5-a)*dt, 'the option price are:')
        print(price_list)
        temp = []
        for a in range(int(len(price_list)/2)):
            temp.append((p*price_list[2*a]+(1-p)*price_list[2*a+1])*np.exp(-r*dt))
        price_list = temp
```

```
M =  5 the lookback option value is : 9.119298985864688
M =  10 the lookback option value is : 10.080582906831015
M =  15 the lookback option value is : 10.519164595672924
```




```
point t =  1.0 the option price are:
, 11.181413117784501, 0.0, 19.452691543130413, 0.0, 9.349916553291678, 6.3745
point t =  0.8 the option price are:
01638813873981, 9.57139153170023, 4.600479677676438, 15.631851880479829, 4.60
point t =  0.6000000000000001 the option price are:
16771005131012, 9.95527127295782, 6.201916453882752, 13.712862965988537, 6.20
point t =  0.4 the option price are:
48076183576446, 9.79911875354703, 7.1479157567747444, 12.168664659721797]
point t =  0.2 the option price are:
27951165547757, 9.504839866450858]
point t =  0.0 the option price are:
1929898586469]
```