```
import numpy as np


S_0 = 100
T = 1
r = 0.08
sigma = 0.2
M_list = [5, 10, 15, 25, 50]
Map = { }

def efficient_lookback(u, d, p, M, N, S, Mx):
    if (S, Mx) in Map:
        return Map[(S, Mx)]

    if N == M:
        Map[(S, Mx)] = Mx-S
        return Mx-S

    U = efficient_lookback(u, d, p, M, N+1, S*u, max(Mx, S*u))
    D = efficient_lookback(u, d, p, M, N+1, S*d, max(Mx, S*d))
    f = (p*U + (1-p)*D)*np.exp(-r*T/M)
    Map[(S, Mx)] = f
    return f

for M in M_list:
    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
    Map.clear()
    value = efficient_lookback(u, d, p, M, 0, S_0, S_0)
    print('For M =  ', M, 'lookback option value is', value)
```

```
⇨  For M =   5 lookback option value is 9.11929898586469
   For M =   10 lookback option value is 10.08058290683101
   For M =   15 lookback option value is 10.519164595672923
   For M =   25 lookback option value is 11.003495335646338
   For M =   50 lookback option value is 11.510862222177268
```