

# American Binomial Hint

April 3, 2017

**\*\* American Binomial Hint \*\***

The sample code below should get your juices flowing about how to model the binomial tree for an American option. The tricky bit is specifying the loops and carefully indexing the indirection necessary to carry out the recursive solution.

The example below comes from page 310 in the McDonald book. See Figure 10.6 for a three-period tree. Although the solution in the figure is for a European put, we can use it below to see how to work with an American option with the same parameters.

Please look at this and we can talk some more in class.

In [2]: `import numpy as np`

```
spot = 41.0
strike = 40.0
rate = 0.08
div = 0.0
vol = 0.30
expiry = 1.0
steps = 3
nodes = steps + 1
dt = expiry / steps
St = np.zeros((nodes, nodes))

u = np.exp((rate - div) * dt + vol * np.sqrt(dt))
d = np.exp((rate - div) * dt - vol * np.sqrt(dt))
pstar = (np.exp(rate - div) - d) / (u - d)

for i in range(nodes):
    St[i, steps] = spot * (u ** (steps - i)) * (d ** i)

for i in range(steps - 1, -1, -1):
    for j in range(i+1):
        St[j, i] = St[j, i+1] / u

print(St)
```

```
[[ 41.          50.07109093  61.14912553  74.6781323 ]
 [  0.          35.4113947   43.24602839  52.81404438]
 [  0.           0.          30.58455792  37.3512727 ]]
```

```
[ 0.          0.          0.          26.41565494]]
```

While using a two-dimensional ndarray is not the most efficient data structure for this problem (and we will replace it later), you should at least see how this could work. You will need to do something similar for the option price tree and derive its values from the spot price tree above.

```
In [ ]:
```