1. Implement a compound Put option, the option to purchase a European-style Put option, with expiry $T$ and strike price $K$, for price $L$ at time $T_1$ satisfying $0 < T_1 < T$. Use your code to price such an option with parameters $(T, T_1, S_0, K, L, r, v, N) = (2, 1, 90, 95, 4.50, 0.03, 0.15, 20)$. (Hint: modify CRRcc.m.)

2. Let $\mathcal{M}(n)$ be the set of path numbers in a recombining binomial tree to depth $n$, as defined in Eq.4.6. Prove that

\[ \mathcal{M}(n + 1) = [2 \mathcal{M}(n)] \cup [2 \mathcal{M}(n) + 1], \]

where $aX + b \overset{\text{def}}{=} \{ax + b : x \in X \}$ for sets $X$ of numbers.

3. Implement floating strike option pricing in the CRR model using geometric means instead of arithmetic means, as in CRRgeo versus CRRaro. Compare the results on the suggested example inputs.

4. Use CRRaro to compute the average-rate Call and Put premiums in the CRR model for $S_0 = K = 100$, $T = 1$, $r = 0.05$, $v = 0.15$, and different values of $N$. Compare $C(0) - P(0)$ with the limit value in Eq.4.20.

5. Implement floating strike option pricing in the CRR model using path-dependent Arrow-Debreu securities. Check that the results agree with CRRflt.

6. Write an Octave program to compute the maximums along all paths in a non-recombining binary tree of depth $N$. (Hint: modify NRTmin().)

7. Implement lookback option pricing in the CRR model using path-dependent Arrow-Debreu securities. Check that the results agree with CRRlb.

8. Implement ladder Put option pricing in the CRR model by modifying CRRladC appropriately. Check that the premium is at least as great as that for the vanilla European-style Put with the same parameters.