Rabbits and Recurrence Relations Problem

This problem can be found at the following link: <http://rosalind.info/problems/fib/>

The purpose of this project is to calculate the number of pairs of rabbits after n number of generations. This project aims at utilising recurrence relationships, and the principle of dynamic programming where the previous element in a sequence is used to calculate the next element.

*To go about solving this problem:*

1. First, I need to calculate the number of rabbit pairs for the first 3-4 generations. This will require a specialised starting algorithm that takes input of the number of pairs found per litter (k-value). This component is important because for a recurrence relationship to work, there need to be prior elements from which the network can build from.
   1. To achieve this, I think I can use two lists. The first list includes the immature rabbits, and the second list includes the mature rabbits.
   2. For the first generation where the one pair are immature, they are found in the immature list. They (all of the elements in the immature list) are then moved to the mature list. To add new immature rabbits, the adding statement can be included right at the end of the loop – this overcomes any new rabbits from being transferred into the mature list right after being added; this is wrong because the rabbits must first mature for a generation before they become mature (and are added to the mature list)
   3. Right after the immature pairs are moved into the mature list, the number of pairs.
   4. In order, the immature rabbits are moved to the mature list. After this, new immature rabbits must be added. The number of immature rabbits added is calculated with this formula: *number of mature rabbits x k-number*.
   5. After the first couple of generations are modelled, I can use the recursion equation I have designed to calculate the number of rabbits in the next generation.
2. After the first few generations are calculated, they can be found using a recurrence relationship to minimise the computing cost when values become very large. For a recurrence relationship to be implemented, a formula is required. I have devised a formula for this relationship, which I will outline later how I came about finding it.

*Recurrence Relationship Formula:*

*Fn = Fn-1 + (mn-1 + (mn-2)2*

*Where:*

*Fn is the number of pairs in the generation you want to find*

*Fn-1 is the number of pairs in the previous generation*

*Mn-1 is the change in pairs from Fn-2 to Fn-1*

*Mn-2 is the change in pairs from Fn-3 to Fn-2*

*How was this formula devised?*

To find this formula for the recurrence relationship, I built up on the knowledge of Fibonacci. I knew that the number of pairs in the next generation, at minimum, contains the number of pairs from the previous generation. However, the rate of change is what changes as the sequence progresses because there are progressively more rabbit pairs, and more of them become mature. To find the rate of change for the formula, I manually found what the number of pairs are for the first 7 generations; analysing the change between generations to see whether I can spot a pattern. The pattern I noticed was that the change from the previous generation to the next was equal to the change for the previous generation, plus the change for the generation before that, squared.

*Interesting ideas to implement:*

I am interested in whether there is any computational benefit in using a recursion to model population growth compared to algorithmic calculation. There must obviously be a difference – but can I prove it? It might require a thousand generations to provide proof. It would also be interesting to see whether the increase in computational time is linear with an increase in generations, or whether it is an exponential graph. This provides an interesting opportunity to use a graphing library in Python to graph the amount of time it takes.