# Rabbits and Recurrence Relations

Orr Shomroni

27 September, 2022

## Problem

A **sequence** is an ordered collection of objects (usually numbers), which are allowed to repeat. Sequences can be finite or infinite. Two examples are the finite sequence $(\pi, -\sqrt{2}, 0, \pi)$ and the infinite sequence of odd numbers $(1,3,5,7,9,\ldots)$. We use the notation $a_n$ to represent the $n$-th term of a sequence.

A **recurrence relation** is a way of defining the terms of a sequence with respect to the values of previous terms. In the case of Fibonacci's rabbits from the introduction, any given month will contain the rabbits that were alive the previous month, plus any new offspring. A key observation is that the number of offspring in any month is equal to the number of rabbits that were alive two months prior. As a result, if $F_n$ represents the number of rabbit pairs alive after the $n$-th month, then we obtain the **Fibonacci sequence** having terms $F_n$ that are defined by the recurrence relation $F_n = F_n - 1 + F_n - 2$ (with $F_1 = F_2 = 1$ to initiate the sequence). Although the sequence bears Fibonacci's name, it was known to Indian mathematicians over two millennia ago.

When finding the $n$-th term of a sequence defined by a recurrence relation, we can simply use the recurrence relation to generate terms for progressively larger values of n. This problem introduces us to the computational technique of **dynamic programming**, which successively builds up solutions by using the answers to smaller cases.

**Given**: Positive integers $n \leq 40$ and $k \leq 5$.

**Return**: The total number of rabbit pairs that will be present after $n$ months, if we begin with 1 pair and in each generation, every pair of reproduction-age rabbits produces a litter of $k$ rabbit pairs (instead of only 1 pair).

## Sample Dataset

```
5 3
```

## Sample Output

```
19
```

```python
def fib(n, k):
  if n<0:
    print("Incorrect input")
  elif n==0:
    return 0
  elif n==1 or n==2:
    return 1
```

```python
    else:
        return fib(n-1,k) + fib(n-2,k)*k

n=33
k=3
F=fib(n,k)
string="The total number of rabbit pairs present after "+str(n)+" months, with each adult\n"+"pair produ
print(string)
```

```
## The total number of rabbit pairs present after 33 months, with each adult
## pair producing 3 pairs of offsprings is 249650241628
```