Rabbits and Recurrence Relations

Jan Emmanuel Samson

2024-07-29

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,…). 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](https://rosalind.info/problems/fib/), 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 rubbits that were live 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 mathematics over two millenia ago.

When finding the n-th term of a sequence defined by recurrence relation, we can simply use the recurrence relation to generate terms for progressively larger values of n. This problem introduces us to the computation technique of **dynamic programming**, which succesively bulds 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

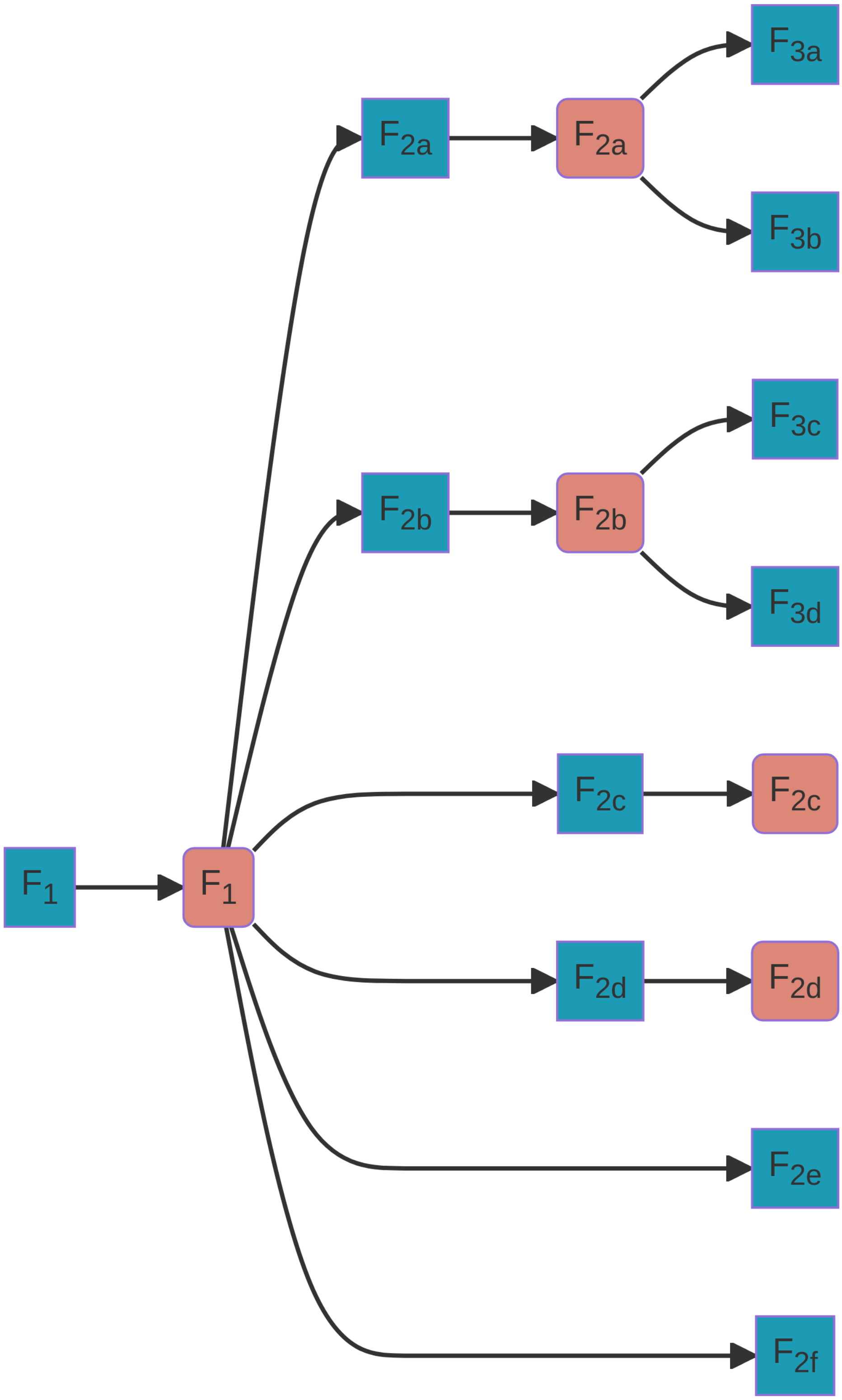
```
5 3
```

Sample Output

```
19
```

Intuition

Express the recurrence relationship using an algebraic equation. In this case, a rabbit pair at reproduction age produces k pairs of offspring. The length of a generation is a single month. The offspring would need to mature an entire generation before producing its own progeny. We can illustrate how a rabbit population (starting with 1 pair) propagates after 5 generations:



Each box represents a rabbit pair where blue corresponds to sexually-immature rabbit pairs and orange to sexually-mature rabbit pairs. Arrows connect parents to their offsprings.

The first two months will always have a single rabbit pair therefore we only update the values after two months. In the third, fourth, and fifth months, the total amount of rabbits can be computed as:

$$p_3 = p_1 + k * p_2$$

$$p_4 = p_2 + k * p_3$$

$$p_5 = p_3 + k * p_4$$

where p_i is the total rabbit pairs in the i^{th} generation. The formula can be generalized as:

$$p_n = p_{n-2} + k * p_{n-1}$$

From the example input, we expect 19 rabbits after 5 months, assuming that each mature pair will produce 3 pairs of offspring per month.

Solution

```
def fibunnyci(n: int, k: int) -> int:
    """Return the number of rabbit pairs after `n` generations."""
    a, b = 1, 1
    for i in range(2, n):
        a, b = b, k*a + b
    return b
```

```
n, k = 5, 3
result = fibunnyci(n, k)
print(result)
```

```
19
```