Subsets of $\{1, 2, ..., n\}$ containing exactly one pair of consecutive integers

Vamshi Jandhyala

October 2024

Contents

1 Problem

Find the number of subsets of $\{1, 2, 3, ..., n\}$ that contain exactly one pair of consecutive integers. Examples of such subsets are $\{\mathbf{1}, \mathbf{2}, 5\}$ and $\{1, 3, \mathbf{6}, \mathbf{7}, 10\}$.

2 Solution using recurrences and generating functions

There are numerous ways to solve this problem but here is my solution involving **generating functions** because I absolutely them.

2.1 Subsets with no consecutive integers

We first find the number of subsets of $\{1, 2, ..., n\}, n \geq 2$ with no consecutive integers.This is equivalent to finding the number of bit strings of length n which don't have consecutive ones. Let $g(n)$ be the number of bit strings with the required property.

If a bit string with the required property starts with one, then the bit immediately next to it cannot be a one. The number of bit strings with the required property that start with a one is $g(n - 2)$.

Similarly, it is easy to see that the number of bit strings with the required property that begin with a zero is $g(n - 1)$.

Therefore, we have $g(n) = g(n - 1) + g(n - 2), g(2) = 3, g(3) = 5$. The form of the recurrence is the same as that of the **Fibonacci sequence**, so $g(n) = F_{n+2}$.

2.2 Subsets with one pair of consecutive integers

Finding the subsets with one pair of consecutive integers is equivalent to finding the number of bit strings with exactly one pair of consecutive ones so we follow the same approach as in the previous section. Let $f(n)$ be the number of strings with the required property.

When a bit string with the required property starts with two consecutive ones, the rest of the string cannot have any other pair of consecutive ones. Therefore, number of bit strings with the required property starting with two consecutive ones or ending in two consecutive ones is $2g(n-3) = 2F_{n-1}$.

A bit string with exactly one pair of consecutive ones in the 'middle' can be formed by taking two bit strings, with the left bit string ending in a one and the right bit string beginning with a one and both strings not having any consecutive ones in them like in the figure below:

0	1	1	0	1

In the above example, we have $n = 5$ where the left string is of length 2 and the right string is of length 3. We see that the total number of bit strings with exactly one pair of consecutive ones in the middle is

$$\sum_{i=2}^{n-2} g(i-2)g(n-i-2) = \sum_{i=2}^{n-2} F_i F_{n-i}. \tag{2.1}$$

Combining the above, we see that the total number of bit strings with exactly one pair of consecutive integers is

$$f(n) = \sum_{i=2}^{n-2} F_i F_{n-i} + 2F_{n-1} = \sum_{i=1}^{n-1} F_i F_{n-i}. \tag{2.2}$$

2.3 Using generating functions to get a closed form solution

We start with the well known generating function of the Fibonacci sequence which is

$$F(x) = \frac{x}{1-x-x^2}. \tag{2.1}$$

We have

$$f(n) = \sum_{i=1}^{n-1} F_i F_{n-i} = [x^n]\frac{x^2}{(1-x-x^2)^2} = [x^{n-2}]\frac{1}{(1-x-x^2)^2} \tag{2.2}$$

where $[x^n]$ denotes the coefficient of x^n.

We can write

$$\frac{1}{(1-x-x^2)^2} = \frac{1}{(r_+ - r_-)^2}\left(\frac{r_+^2}{(1-xr_+)^2} + \frac{r_-^2}{(1-xr_-)^2} - \frac{2r_+r_-}{1-x-x^2}\right) \tag{2.3}$$

where $r_\pm = \frac{1\pm\sqrt{5}}{2}$.

Therefore

$$\begin{aligned} f(n) = [x^{n-2}]\frac{1}{(1-x-x^2)^2} &= \frac{1}{5}\left((n-1)r_+^n + (n-1)r_-^n + 2F_{n-1}\right) \\ &= \boldsymbol{\frac{n-1}{5}L_n + \frac{2}{5}F_{n-1}} \end{aligned} \tag{2.4}$$

where L_n is the n^{th} **Lucas number**.

When $n = 10, f(10) = \frac{9\cdot 123}{5} + \frac{2\cdot 34}{5} = \mathbf{235}$.

3 Computational Solution

From the following Python code we see that the number of subsets with exactly one pair of consecutive integers is **235**.

```python
def has_exactly_one_consecutive_pair(subset):
    return sum(subset[i] == '1' and subset[i+1] == '1' for i in range(len(subset) - 1)) == 1

def count_subsets_with_one_consecutive_pair(n):
    count_by_size = {i: 0 for i in range(1, n+1)}
    total_count = 0

    for i in range(1, 2**n):
        subset = format(i, f'0{n}b')
        if has_exactly_one_consecutive_pair(subset):
            size = subset.count('1')
            count_by_size[size] += 1
            total_count += 1

    return total_count, count_by_size

n = 10
total, breakdown = count_subsets_with_one_consecutive_pair(n)
print(f"Total number of subsets with exactly one consecutive pair: {total}")
```