Counting triangles

Examen MOSIG1 2020

We are interested in this problem in solving a classical combinatorial puzzle.

We consider a series of large triangles, denoted by T_k, that are composed of smaller isosceles triangles (called the basic triangles) whose length of the three sides are equal to 1. The series is built by adding at the bottom of the current large triangle of rank k a row of basic triangles as shown in figure 1. The game consists in the enumeration of all the triangles that compose the large ones (including the basic ones and all the other upper-sized triangles).

In other words, we want to determine how many triangles are contained into T_k for all $k \geq 1$. Let us denote this number by N_k.

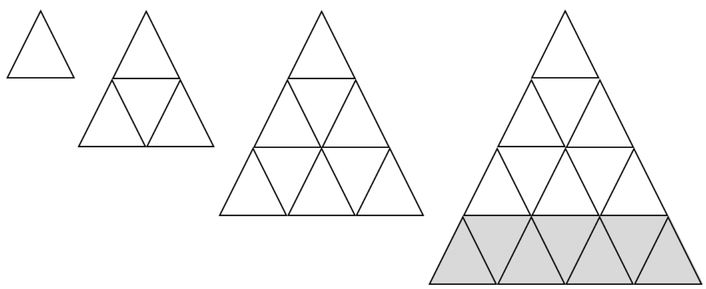

Figure 1: The 4 first terms of the progression of the T_k (from left to right). Each one is built by adding a row of basic triangles on the basis of the previous one (as shown for T_4).

Let us determine the first elements of the progression:

1. Of course, the first one (T_1) contains only one basic triangle, thus, $N_1 = 1$.
2. We have two types of triangles in the second triangle of the progression (T_2), namely a large one whose side is equal to 2 and 4 basic triangles, thus $N_2 = 1 + 4 = 5$.
3. Similarly, there are 3 types of triangles contained into T_3: A large one of side 3, 3 medium triangles of side 2 and 9 basic ones, then $N_3 = 1 + 3 + 9 = 13$.

QUESTION 1. Determine the number of triangles contained into the fourth term of the progression T_4.

You can proceed as before by the enumeration of triangles by increasing sizes and determine each $N_4^{(i)}$ for $1 \leq i \leq 4$.

Let us now generalize to any values of k.

QUESTION 2. Determine the recurrence equation for computing T_k knowing T_{k-1}, T_{k-2} and T_{k-3}.

We ask here for a detailed and argumented answer.

The expression is as follows:
For even k $N_k = 3(N_{k-1} - N_{k-2}) + N_{k-3} + 1 + 1$
For odd k : $N_k = 3(N_{k-1} - N_{k-2}) + N_{k-3} + 1$
where $k \geq 3$ and $N_0 = 0$, $N_1 = 1$ et $N_2 = 5$.

We turn now to the problem of computing *efficiently* the kth term of this progression.

QUESTION 3. Compute the 8 first values of the T_k.

We can compute them directly by the previous expression or using a trick that considers the new derived progression: $\Delta_k = N_{k+1} - N_k$, and then, again by computing the difference $\Gamma_k = \Delta_{k+1} - \Delta_k$ and finally, $\Gamma_{k+1} - \Gamma_k$.

Draw the corresponding table whose first row is composed by $k = 1, 2, 3, \ldots$, the second row by the Δ_k and the third one by the Γ_k and the fourth one by the differences. Then, derive a low-cost method that allows to fill the table for any k.

This method is much more efficient than the direct computations using the recurrence equations, however, it can even be used further for deriving a close formula (which means: being able to compute directly the value of N_k for any k).

The process is to concentrate to the last row of the previous table and focus on the even k. The values of the previous row (the one with the Γ) can be obtained by simple polynomials of degree 1.

QUESTION 4. Determine this polynomial in k.

We can proceed similarly for the previous row (the one corresponding to the Δ aiming a polynomial of degree 2 end then again for the N_k with polynomial of degree 3.

QUESTION 5. Determine these polynomials and conclude.

$N_k = \frac{k.(k+2).(2k+1)}{8}$ if k is even and $N_k = \frac{k.(k+2).(2k+1)-1}{8}$ if k is odd.