

A bouquet of counting problems

1 Election \pm problem

Consider a closely fought election in West Bengal on April 2026 between two major political alliances, say Party A and Party B . As the counting of votes progresses constituency by constituency, television channels continuously display the running totals. Sometimes one party leads early, then the other catches up, and the lead may fluctuate dramatically throughout the day.

Now suppose Party A eventually wins by a very small margin: it receives $n + 1$ constituencies while Party B receives n . A natural combinatorial question arises: In how many possible counting sequences does Party A remain always ahead of (or at least tied with) Party B during the entire counting process?

Example. For $n = 2$, there are five such sequences:

$AAABB$, $AABAB$, $AABBA$, $ABAAB$, $ABABA$.

Exercise 1.1:

Find out the number of such sequences for $n = 3, 4, 5$.

2 Well-formed formulas

Suppose p and q are variables and $*$ is a binary operation (you can think of addition or multiplication, for example). Here are some logical syntactic rules to form new expressions using the symbols $p, q, *$ and parentheses—the so-called *well-formed formulas*:

1. Every variable is a well-formed formula.
2. If s and t are well-formed formulas, then so is $(s * t)$.

Some well-formed formulas include $p, q, (p * q), ((p * p) * q)$ etc. Quick tests to determine if a given string of symbols is not a well-formed formula are the following.

Exercise 2.1:

1. The number of left parentheses '(' in a well-formed formula equals the number of right parentheses ')
2. (**Strong prefix property**) If s is a well-formed formula that is not a variable, and t is a proper left substring of s containing at least one symbol, then the number of left parentheses in t is strictly greater than the number of right parentheses in it.

Suppose you are tasked with arranging n pairs of parentheses while satisfying the **weak prefix property**: the number of left parentheses in any proper left substring is at least the number of right parentheses in it. For $n = 1$, you have only one choice, namely $()$ but for $n = 2$, you have two choices, namely $()()$ or $(())$.

Exercise 2.2:

Find all expressions for $n = 3, 4, 5$ satisfying the weak prefix property. Can you relate this problem to Exercise 1.1?

Exercise 2.3:

Let C_n denote the number you counted in the above exercise. Is it possible to compute C_{n+1} if you know C_0, C_1, \dots, C_n ?

3 Building Mountains

Did you know that the tallest mountain in the world is Mauna Kea in Hawaii, and not Mount Everest, when measured from its base to peak?

Suppose you want to build a mountain range using n up-strokes ($/$) and n down-strokes (\backslash) but you can never go below the sea level.

For $n = 1$, you have only one choice: \wedge . For $n = 2$, you can go $\wedge\wedge$ or $/\wedge\backslash$. For $n = 3$, the landscape becomes more varied. You might build three small peaks, or one large peak with a smaller one along with it.

Exercise 3.1:

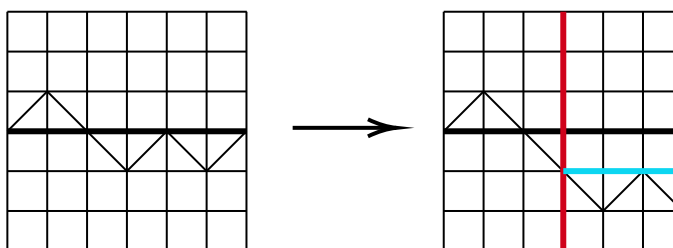
List and count the distinct mountain ranges that you can build for $n = 5$.

Counting mountain ranges using reflection (André's Method)

Let M_n denote the number of mountain ranges constructed using n -many of $/$ and n -many \backslash . The counting of M_n proceeds as follows: we will subtract the number of bad mountain ranges from the number of all possible mountain ranges, which can go below the sea level as well.

Step 1. What is the total number of mountain ranges using exactly n -many $/$ and exactly n -many \backslash ?

Step 2. The Reflection: Look at the first instance where the mountain range drops below the sea level. We will reflect (flip $/$ and \backslash) the mountain range after that, as shown in the following figure.



Step 3. Note that the new mountain range has $(n-1)$ -many / and $(n+1)$ -many \. The reverse process of Step 2 can be applied to any mountain range having $(n-1)$ -many / and $(n+1)$ -many \ to get a bad mountain range.

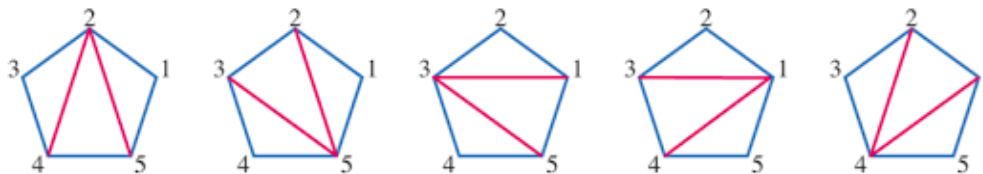
Step 4. What is the total number of mountain ranges having $(n-1)$ -many / and $(n+1)$ -many \?

Step 5. Subtract the number obtained in Step 1 from the number obtained in Step 4 to get the expression for M_n .

4 Triangulations of Polygons

Suppose you are lost in the Amazon forest but are holding a GPS-enabled device. Even if the forest contains cell towers only on its boundary, your location is determined by pinging three closest cell towers and calculating your distance from them—this method is called *triangulation*.

To understand the concept of triangulation mathematically, imagine a regular $(n+2)$ -gon with vertices labeled 1 through $(n+2)$. Divide this polygon into exactly n triangles using $n-1$ non-intersecting diagonals. For a pentagon ($n=3$), there are exactly five ways to achieve this.



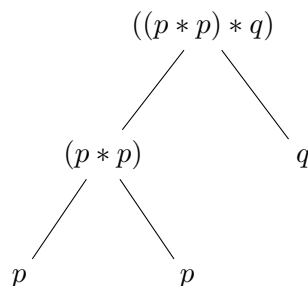
Exercise 4.1:

Count the number of triangulations of a (vertex-labeled) 6-gon and 7-gon. Is this problem related to Exercise 3.1? (If you find this second problem hard, you could revisit it after completing the next section.)

5 Full Binary Trees

Full binary trees are used by computers in the evaluation of well-formed formulas from Section 2—they are rooted trees where every node has either zero or two children. A node with zero children is called a *leaf node* and a node that is not a child of any other node is called the *root node*. An *internal node* is a node that is not a leaf node.

The following picture shows how the evaluation of $((p * p) * q)$ works.



Exercise 5.1:

Count the number of full binary trees with n internal nodes for $n = 3, 4$. Is this problem related to Exercise 2.2?

Exercise 5.2:

Is it possible to view the internal nodes of a full binary tree as triangles in a triangulation of an appropriate regular polygon? (The root could be the unique triangle containing the edge 12 in a given triangulation of the polygon.)

6 Bonus: Generating functions

Recall the numbers C_n from Exercise 2.3. Define the *generating function* of this sequence to be the expression $F(x) := \sum_{n \geq 0} C_n x^n$.

Simple manipulations of such expressions will actually turn out to be an easy way to evaluate sequences that satisfy certain recurrence relations. Here are some rules that will help us.

Exercise 6.1:

Let $A(x) = \sum_{n \geq 0} a_n x^n$. Fix a positive integer m .

1. Express $\sum_{n \geq m} a_{n-m} x^n$ in terms of $A(x)$.
2. Express $\sum_{n \geq 0} a_{n+m} x^n$ in terms of $A(x)$.

Exercise 6.2:

1. Let $A(x) = \sum_{n \geq 0} a_n x^n$ and $B(x) = \sum_{n \geq 0} b_n x^n$. Express $D(x) := \sum_{n \geq 0} (\sum_{k=0}^n a_k b_{n-k}) x^n$ in terms of $A(x)$ and $B(x)$.
2. Let $A_1(x), \dots, A_m(x)$ be defined as $A_i(x) := \sum_{n \geq 0} a_{i,n} x^n$. Express $D(x) := \sum_{n \geq 0} (\sum_{k_1 + \dots + k_m = n} a_{1,k_1} a_{2,k_2} \cdots a_{m,k_m}) x^n$ in terms of $A_1(x), \dots, A_m(x)$.

Exercise 6.3:

Use the recurrence relation obtained in Exercise 2.3 and the rules for manipulating generating functions that we saw above, show that $F(x) = 1 + xF(x)^2$. Use this to find an expression for $F(x)$ in terms of x .