USA Mathematical Talent Search
Solutions to Problem 4/4/18
www.usamts.org

4/4/18. We are given a $2 \times n$ array of nodes, where n is a positive integer. A *valid connection* of the array is the addition of 1-unit-long horizontal and vertical edges between nodes, such that each node is connected to every other node via the edges, and there are no loops of any size. We give some examples for $n = 3$:

Valid   Valid   Invalid: loop   Invalid: not connected

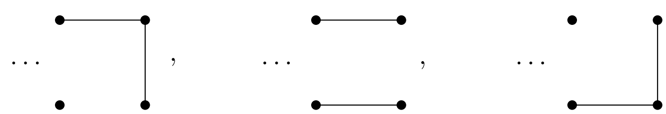
Let T_n denote the number of valid connections of the $2 \times n$ array. Find T_{10}.

Credit This problem was proposed by Naoki Sato.

Comments By constructing valid connections on a $2 \times n$ array from smaller arrays, we can obtain a recursive formula for T_n. *Solutions edited by Naoki Sato.*

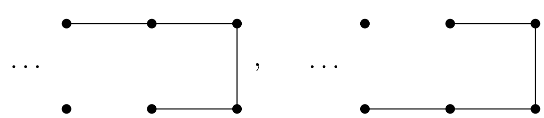
Solution 1 by: Drew Haven (11/CA)

It is trivial to note that $T_1 = 1$ because there is only one way to connect two nodes. Let us compute T_{n+1} from T_n. Given any valid connection of $2 \times n$ nodes, adding two nodes to the right gives an array of size $2 \times (n+1)$. The additional two nodes may be connected in one of three ways:

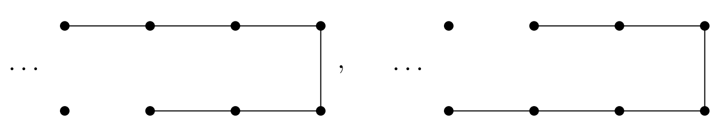


None of these result in any loops. This gives a total of $3T_n$ valid connections.

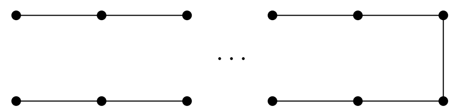
However, it is possible that a valid connection on a $2 \times (n+1)$ array does not contain a valid connection in its leftmost $2 \times n$ nodes. Let us consider the case where the leftmost $2 \times (n-1)$ nodes form a valid connection, but the leftmost $2 \times n$ nodes do not. There are two different ways to connect the nodes on the right to make a valid connection on a $2 \times (n+1)$ array:



This adds $2T_{n-1}$ ways to the total count. Similarly, if only the leftmost $2 \times (n-2)$ nodes make a valid connection, there are $2T_{n-2}$ ways:



Likewise, there are $2T_k$ ways for each $k < n$ that come from the leftmost $2 \times k$ nodes forming a valid connection. The last case to consider is the case when the leftmost two nodes are not connected, and there are no valid connections of any $2 \times k$ leftmost subarray up to $k = n$. There is only one such valid connection:



Summing these ways gives a formula for T_{n+1}:

$$T_{n+1} = 3T_n + 2T_{n-1} + 2T_{n-2} + \cdots + 2T_1 + 1. \tag{1}$$

To find a simpler recurrence, we subtract

$$T_n = 3T_{n-1} + 2T_{n-2} + \cdots + 2T_1 + 1$$

from this to give

$$\begin{aligned} T_{n+1} - T_n &= (3T_n + 2T_{n-1} + 2T_{n-2} + \cdots + 2T_1 + 1) \\ &\quad - (3T_{n-1} + 2T_{n-2} + \cdots + 2T_1 + 1) \\ &= 3T_n - T_{n-1} \\ \Rightarrow \quad T_{n+1} &= 4T_n - T_{n-1}, \end{aligned}$$

which can be rewritten as

$$T_{n+2} = 4T_{n+1} - T_n. \tag{2}$$

From (1), $T_2 = 3T_1 + 1 = 4$. Then from (2),

$$\begin{aligned} T_3 &= 4 \cdot 4 - 1 = 15, \\ T_4 &= 4 \cdot 15 - 4 = 56, \\ T_5 &= 4 \cdot 56 - 15 = 209, \\ T_6 &= 4 \cdot 209 - 56 = 780, \\ T_7 &= 4 \cdot 780 - 209 = 2911, \\ T_8 &= 4 \cdot 2911 - 780 = 10864, \\ T_9 &= 4 \cdot 10864 - 2911 = 40545, \\ T_{10} &= 4 \cdot 40545 - 10864 = 151316. \end{aligned}$$

As a side note, an explicit formula for T_n can be found using generating functions:

$$T_n = \frac{(2+\sqrt{3})^n - (2-\sqrt{3})^n}{2\sqrt{3}}. \tag{3}$$

Substituting 10 for n here gives the same answer, $T_{10} = 151316$.