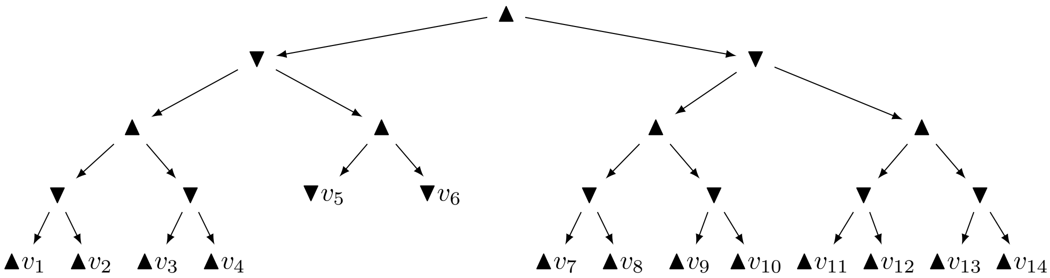
Leaf nodes take values which are either 0 or 1. Assume that we iterate over nodes from left to right. Write down the values of the leaf nodes where the **minimum** / **maximum** number of leaf nodes get skipped.



ANSWERS:

- Each row refers to one accepted combination. This is uniquely determined by v_1 and v_2.
- `X` stands for “don’t care”.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}
Minimum: 0 leaf nodes	0 0 1	0 1 0	1	1	0	0	1	0	1	1	1	0	1	X
Maximum: 8 leaf nodes	0 0 1 1	0 1 0 1	0 0 0 X	X	X X X 1	X	0 0 0 X	X	0 0 0 X	X	X	X	X	X

EXPLANATION:

- We aim to not prune any leaf nodes to attain the minimum. This means that we are aiming for $\alpha = 0$ and $\beta = 1$. The range starts off with $(-\infty, \infty)$.
  - There is a MAX node changing α after we visit v_1 and v_2. We want $\alpha = 0$, so one of the ways to attain that is to set $v_1 = v_2 = 0$. (As long as $v_1 = 0$ or $v_2 = 0$, we’re good.)
  - We now consider the MIN node with v_3 and v_4. β is updated, so we should set $v_3 = 1$.
  - After visiting v_4, we will return all the way back up to the MIN node connecting v_1 to v_6. β is updated, so we should set $v_4 = 1$.
  - Now we descend to the MAX node connecting v_5 and v_6. α is updated, so we set $v_5 = 0$.
  - After visiting v_6, we will return all the way back to the root (which is a MAX node). α is updated, so we should set $v_6 = 0$.
  - Using the same approach, we can decide the values for $v_7, v_8, \ldots, v_{13}$. v_{14} can be either 0 or 1 (that essentially decides the return value of the minimax algorithm).
- For the maximum number of nodes pruned, note that $\alpha = \beta$ is already sufficient to trigger pruning. Therefore, one can set all values to be the same, such that all branches with both α and β set (i.e. not $\pm\infty$) would be pruned.