

La práctica consiste en contar el número de nodos hoja examinados por MiniMax y AlphaBeta para varios tableros del 3-en-rama.

Salvo error, tras completar la práctica la salida del programa `sesionJuegos.TicTacToeDemo.java` debería ser la siguiente:

Mi Demo

MI DEMO:

```
- - -  
- X -  
- - -  
Hojas MiniMax : 25872  
Hojas AlphaBeta: 973
```

```
- - -  
O X -  
- - -  
Hojas MiniMax : 3270  
Hojas AlphaBeta: 145
```

```
- - X  
O X -  
- - -  
Hojas MiniMax : 417  
Hojas AlphaBeta: 53
```

```
- - X  
O X -  
O - -  
Hojas MiniMax : 80  
Hojas AlphaBeta: 12
```

```
X - X  
O X -  
O - -  
Hojas MiniMax : 18  
Hojas AlphaBeta: 8
```

```
X O X  
O X -  
O - -  
Hojas MiniMax : 5  
Hojas AlphaBeta: 5
```

```
X O X  
O X -  
O - X  
Hojas MiniMax : 1  
Hojas AlphaBeta: 1  
Mi DEMO done
```

El número de nodos hoja examinados por alfa-beta es mucho menor en la mayoría de casos.

Tal como se ha comentado en clase una posible solución para la función `maxValueLeafs` sería la que aparece a continuación. La idea básica es que, además del valor minimax de cada nodo (necesario para realizar las podas correspondientes), hay que devolver el número de nodos hoja visitados. Como ambos valores son enteros, se pueden devolver en un array (otros mecanismos son posibles, como por ejemplo una lista):

```
protected int[] maxValueLeaves(GameState state, AlphaBeta ab) {
//esta función debe devolver dos resultados enteros, los guardamos en un array
int[] result = new int[2];
result[0] = Integer.MIN_VALUE; //valor minimax
result[1] = 0; //contador de hojas visitadas

if (terminalTest(state)) {
    result[0] = computeUtility(state);
    result[1] = 1;
    return result;
} else {
    List<GameState> successorList = getSuccessorStates(state);
    for (int i = 0; i < successorList.size(); i++) {
        GameState successor = (GameState) successorList.get(i);
        int[] resSuc = minValueLeaves(successor, ab.copy());
        int minimumValueOfSuccessor = resSuc[0]; //valor minimax del suc.
        int hojasSuc = resSuc[1]; //numero de hojas

        result[1] = result[1] + hojasSuc;
        if (minimumValueOfSuccessor > result[0]) {
            result[0] = minimumValueOfSuccessor;
            state.put("next", successor); //esto no es imprescindible
        }
        if (result[0] >= ab.beta()) {
            return result; //PODA: no se examinan más sucesores
        }
        ab.setAlpha(Util.max(ab.alpha(), result[0])); //actualiza alfa
    }
    return result;
}
}
```

La función `minValueLeafs` es análoga.

NOTA LINGÜÍSTICA: El plural de Leaf (hoja, en inglés) es Leaves, y no Leafs.