**Serpientes y Escaleras**

# Caso 1: Caso Base

En este caso, probamos que el algoritmo funciona según lo esperado

|  |  |
| --- | --- |
| 00.in | 00.out |
| 1  3 19  1  4 16 | 3 |

# Caso 2: Sin Escaleras

En este caso, probamos que el algoritmo pueda reconocer que el tablero no ofrece ninguna ventaja

|  |  |
| --- | --- |
| 01.in | 01.out |
| 1  3 19  0 | 5 |

# Caso 3: Sin Snakes

En este caso, probamos que el algoritmo pueda reconocer que el tablero tiene ventajas significativas

|  |  |
| --- | --- |
| 02.in | 02.out |
| 0  1  4 16 | 3 |

# Caso 4: Snake en mínimo sin escaleras

En este caso, pruebo que el algoritmo puede sacar el mínimo, aun sin escaleras

|  |  |
| --- | --- |
| 03.in | 03.out |
| 1  3 18  0 | 5 |

# Caso 5: Snake en la cima de la escalera

En este caso, la serpiente esta al final de una escalera, lo que obliga a caerse nuevamente

|  |  |
| --- | --- |
| 04.in | 04.out |
| 1  2 12  1  7 12 | 5 |

# Caso 6: Múltiples escaleras

En este caso, pruebo que el algoritmo es capaz de aprovechar las ventajas de multiples escaleras

|  |  |
| --- | --- |
| 05.in | 05.out |
| 1  2 7  2  8 13  13 25 | 2 |

# Caso 7: Tres seguidas

En este caso, pruebo que el algoritmo es capaz de aprovechar las ventajas de multiples escaleras

|  |  |
| --- | --- |
| 06.in | 06.out |
| 1  2 7  3  3 7  7 9  9 18 | 2 |

# Caso 8: Derrotando las serpientes

En este caso, pruebo que el algoritmo es capaz de avanzar por más de tener muchas serpientes seguidas

|  |  |
| --- | --- |
| 07.in | 07.out |
| 5  3 9  9 11  11 18  18 21  21 24  2  8 13  13 22 | 3 |