1.2 What is a Combinatorial Game? We now define the notion of a combinatorial game more precisely. It is a game that satisfies the following conditions.

(1) *There are two players.*

(2) *There is a set, usually finite, of possible positions of the game.*

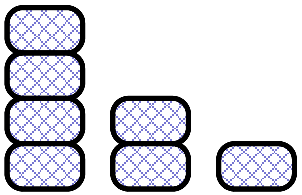
(3) *The rules of the game specify for both players and each position which moves to other positions are legal moves. If the rules make no distinction between the players, that is if both players have the same options of moving from each position, the game is called* **impartial**; *otherwise, the game is called* **partizan**.

(4) *The players alternate moving.*

(5) *The game ends when a position is reached from which no moves are possible for the player whose turn it is to move. Under the* **normal play rule**, *the last player to move wins. Under the* **misère play rule** *the last player to move loses.*

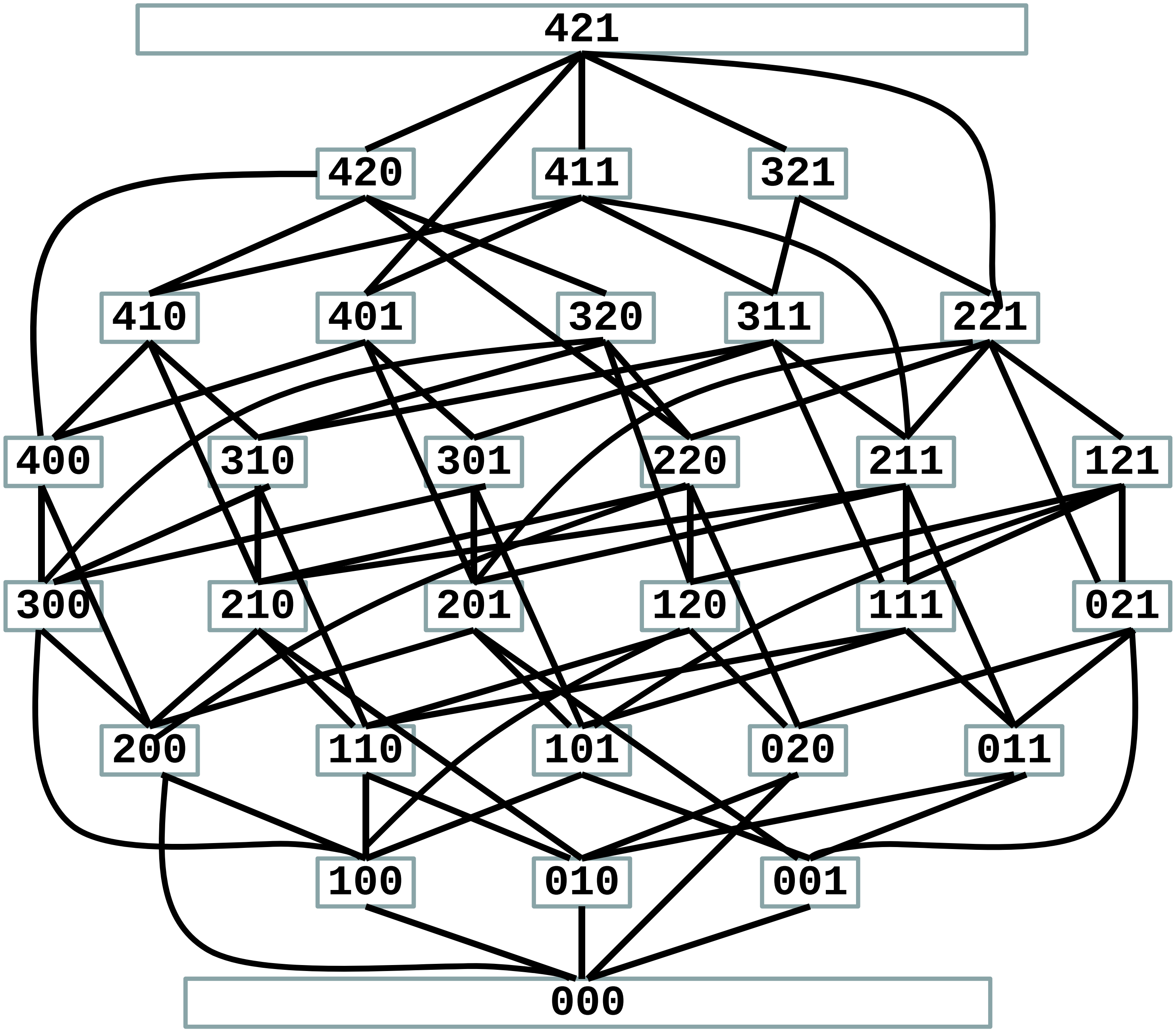
If the game never ends, it is declared a draw. However, we shall nearly always add the following condition, called **the Ending Condition**. This eliminates the possibility of a draw.

(6) *The game ends in a finite number of moves no matter how it is played.*



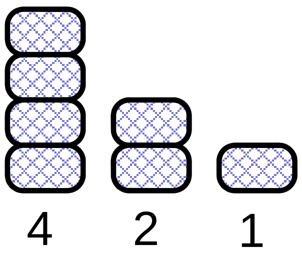
Each player can remove 1 or 2 pegs.

Player who removes the last peg wins.



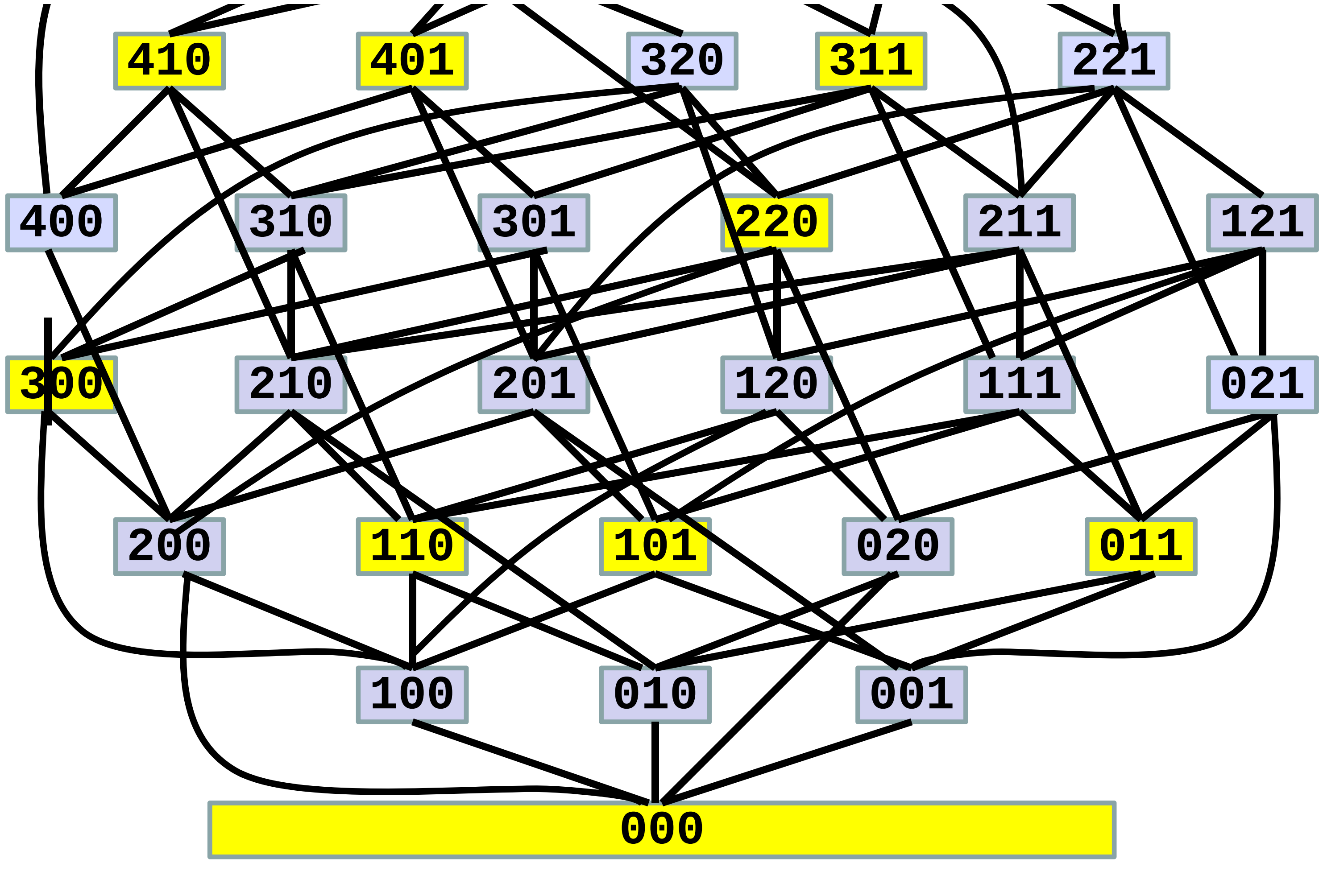
Characteristic Property. *P-positions and N-positions are defined recursively by the following three statements.*

(1) *All terminal positions are P-positions.*
(2) *From every N-position, there is at least one move to a P-position.*
(3) *From every P-position, every move is to an N-position.*



Each player can remove 1 or 2 pegs.

Player who removes the last peg wins.



Determining P and N positions

Step 1: Label every terminal position as a P-position.

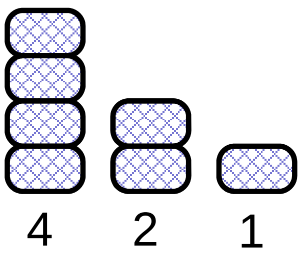
Step 2: Label every position that can reach a labelled P-position in one move as an N-position.

Step 3: Find those positions whose only moves are to labelled N-positions; label such positions as P-positions.

Step 4: If no new P-positions were found in step 3, stop; otherwise return to step 2.

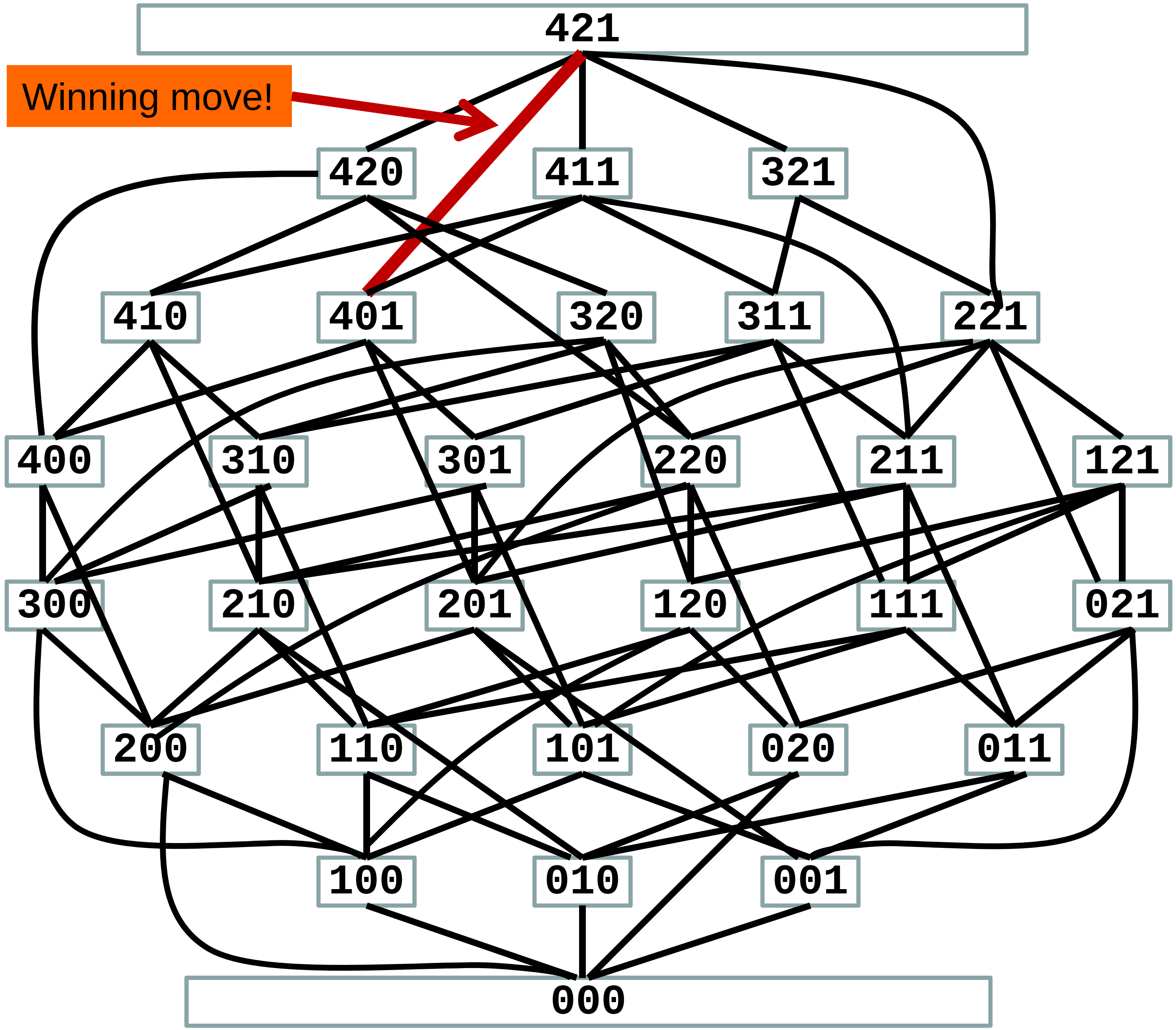
It is easy to see that the strategy of moving to P-positions wins. From a P-position, your opponent can move only to an N-position (3). Then you may move back to a P-position (2). Eventually the game ends at a terminal position and since this is a P-position, you win (1).

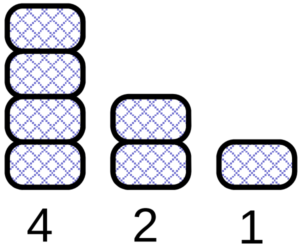
Here is a characterization of P-positions and N-positions that is valid for impartial combinatorial games satisfying the ending condition, under the normal play rule.



Each player can remove 1 or 2 pegs.

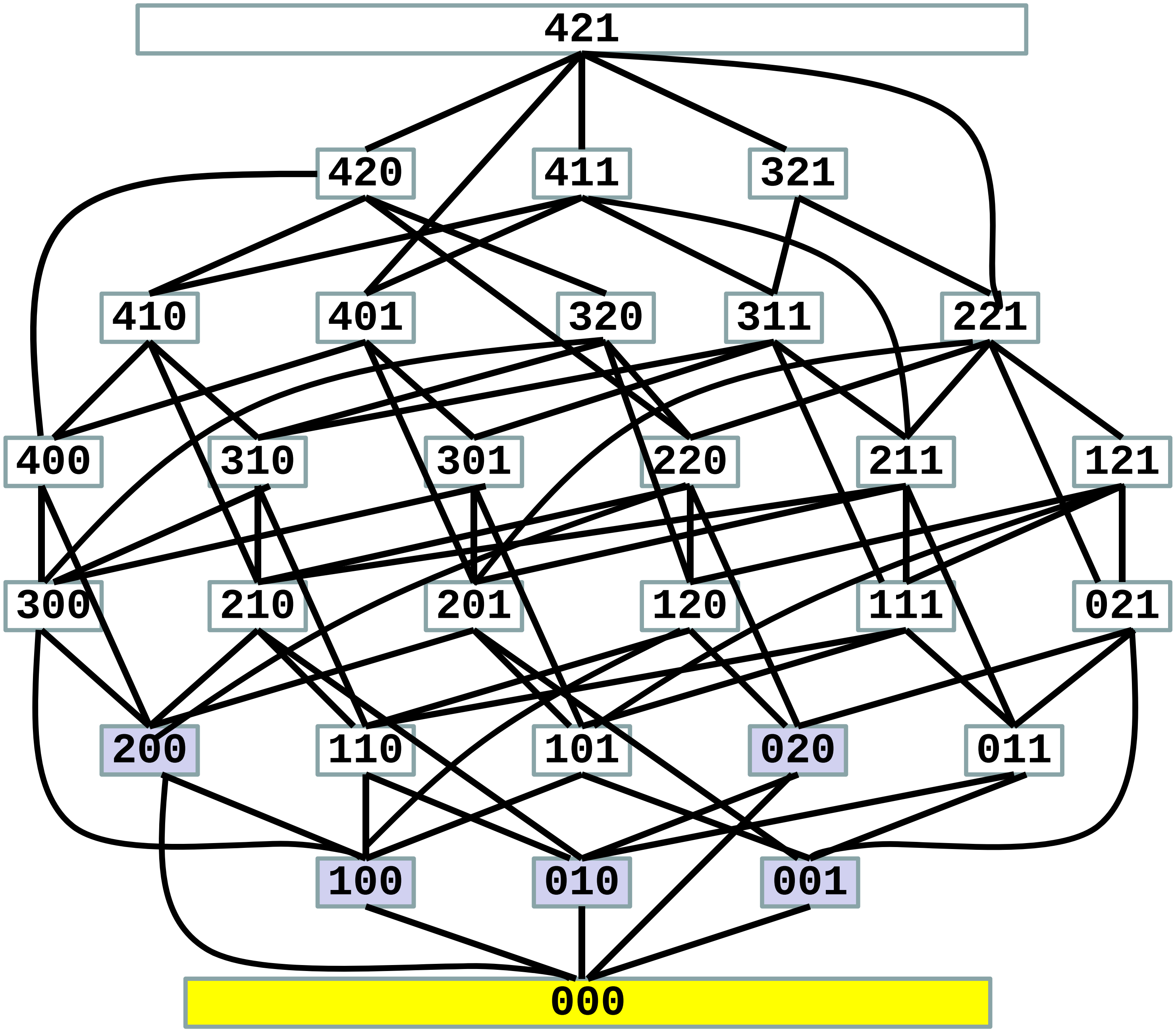
Player who removes the last peg wins.

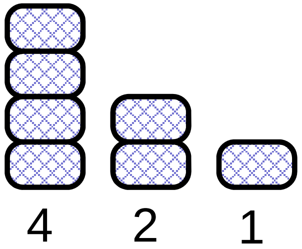




Each player can remove 1 or 2 pegs.

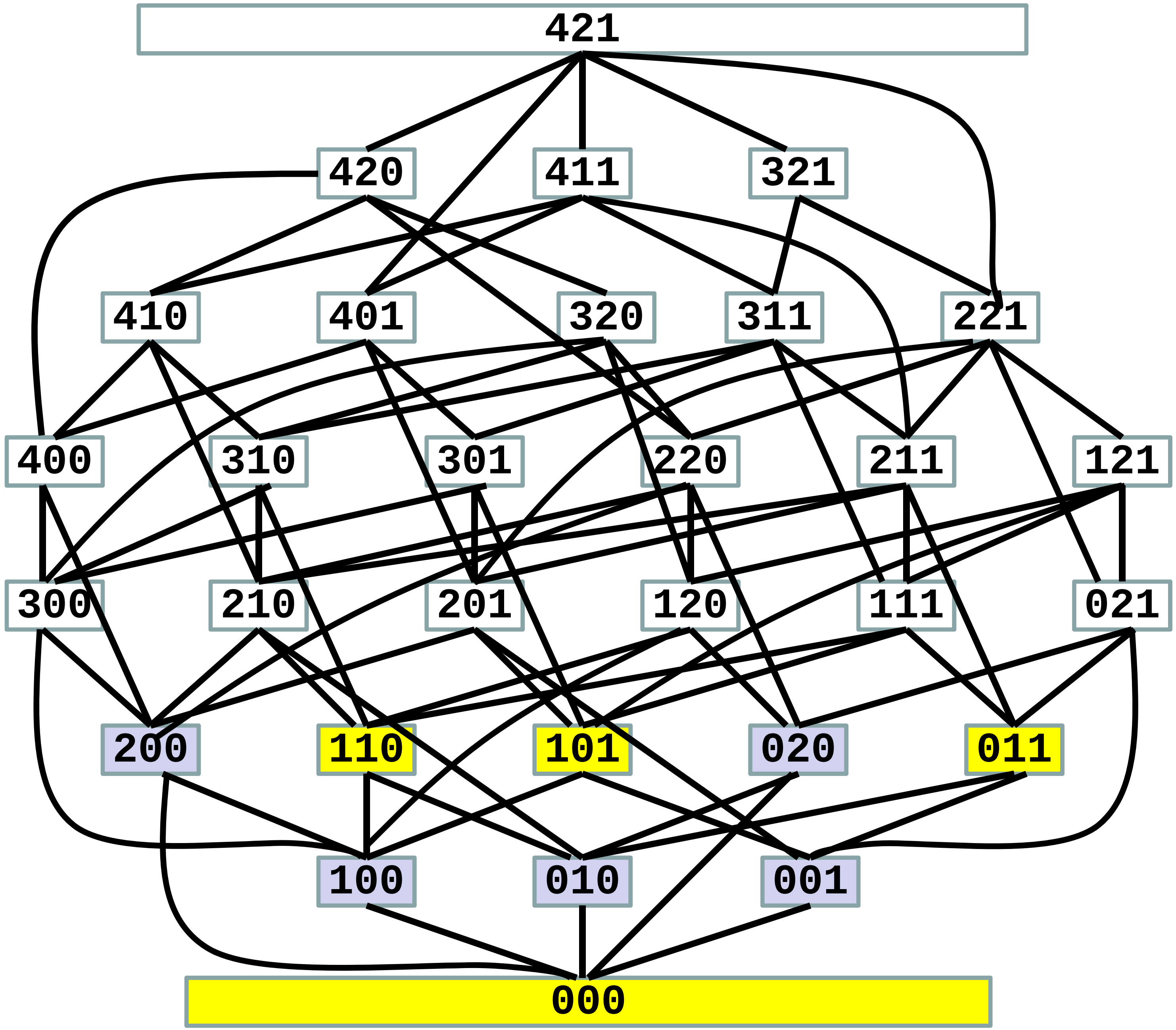
Player who removes the last peg wins.

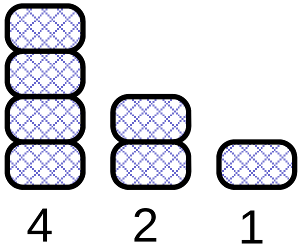


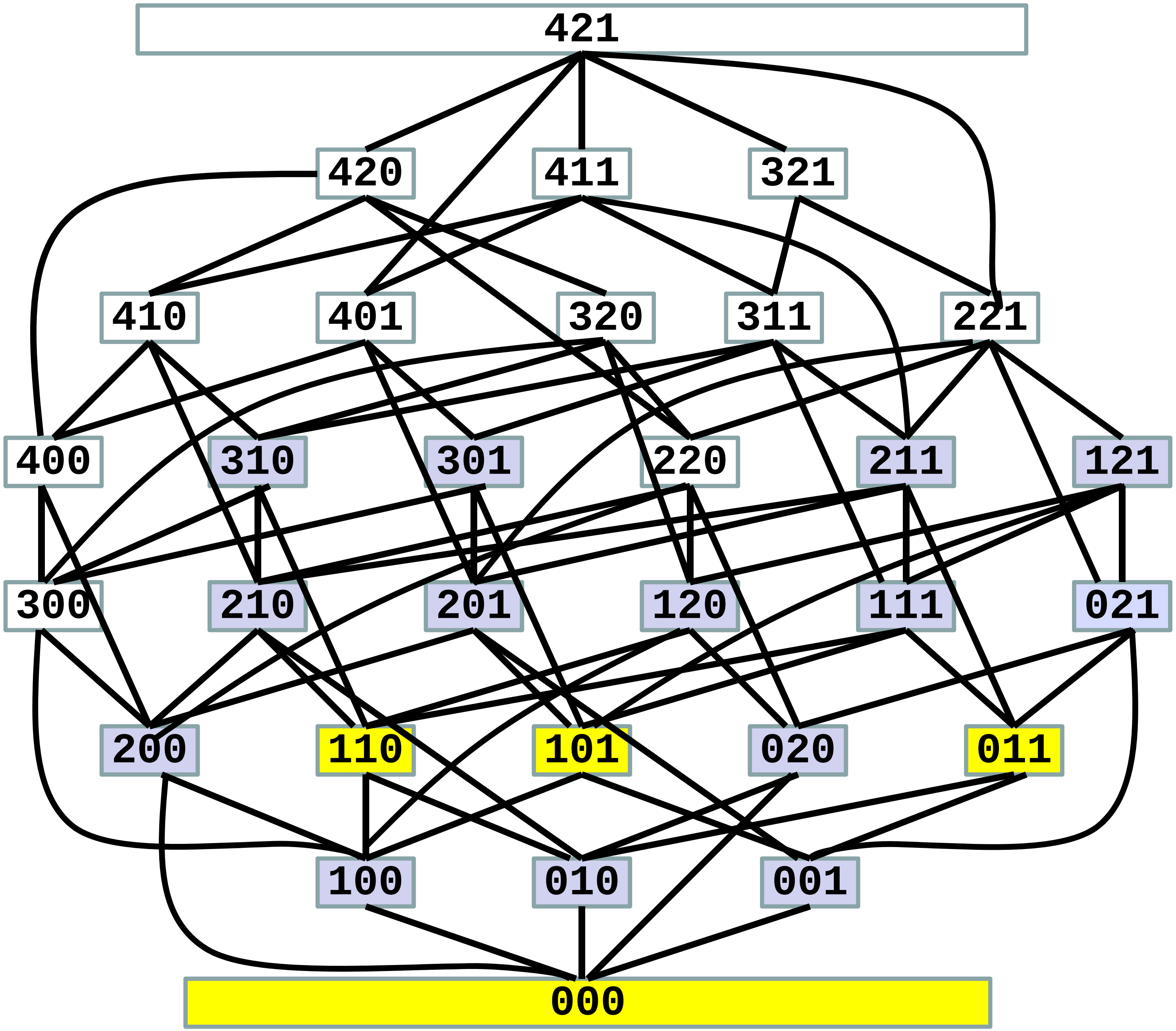


Each player can remove 1 or 2 pegs.

Player who removes the last peg wins.

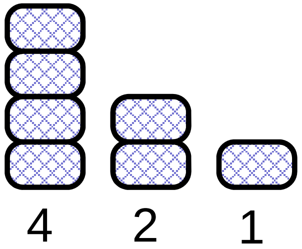






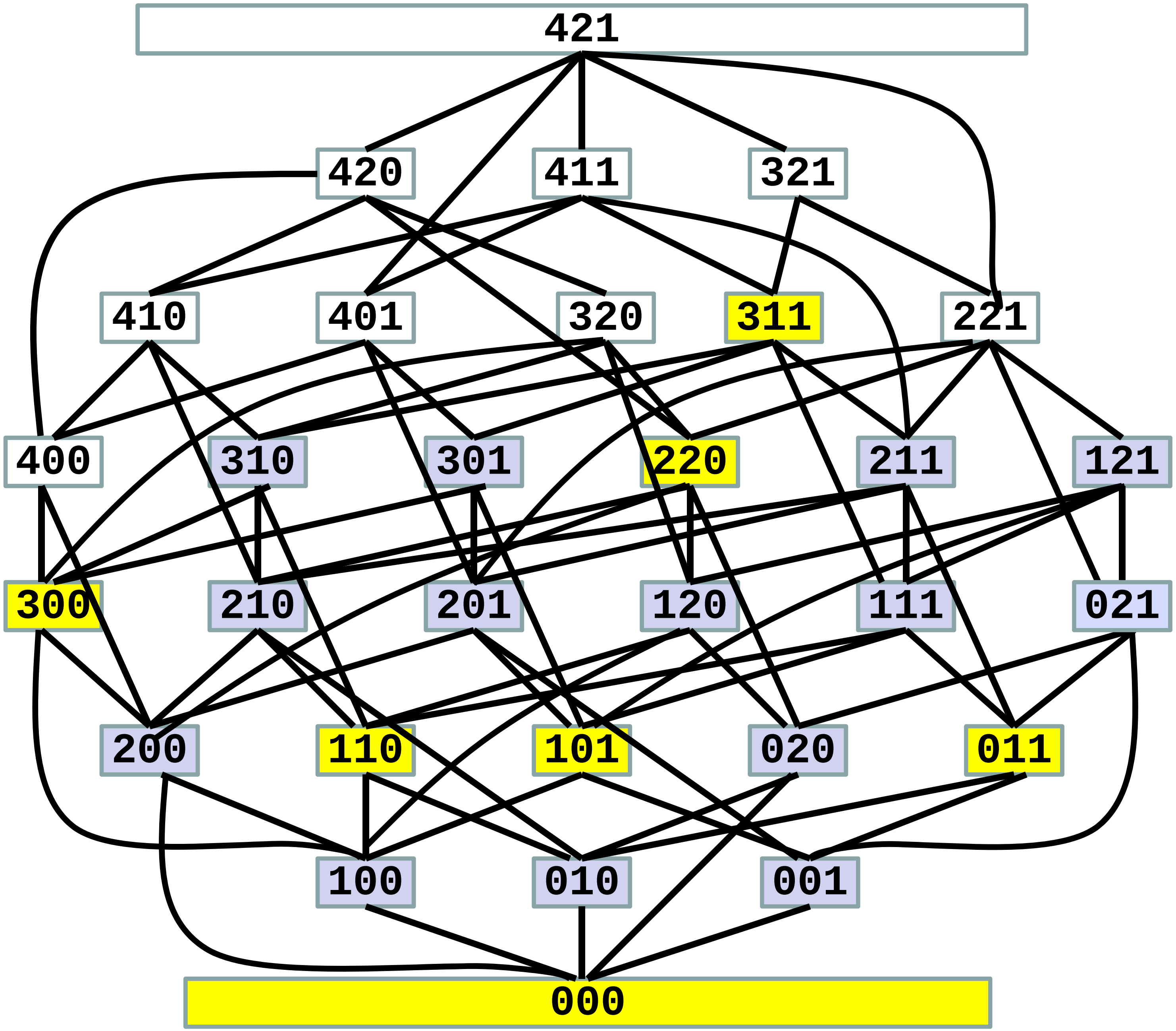
Each player can remove 1 or 2 pegs.

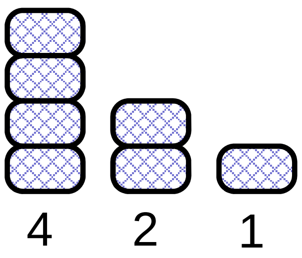
Player who removes the last peg wins.



Each player can remove 1 or 2 pegs.

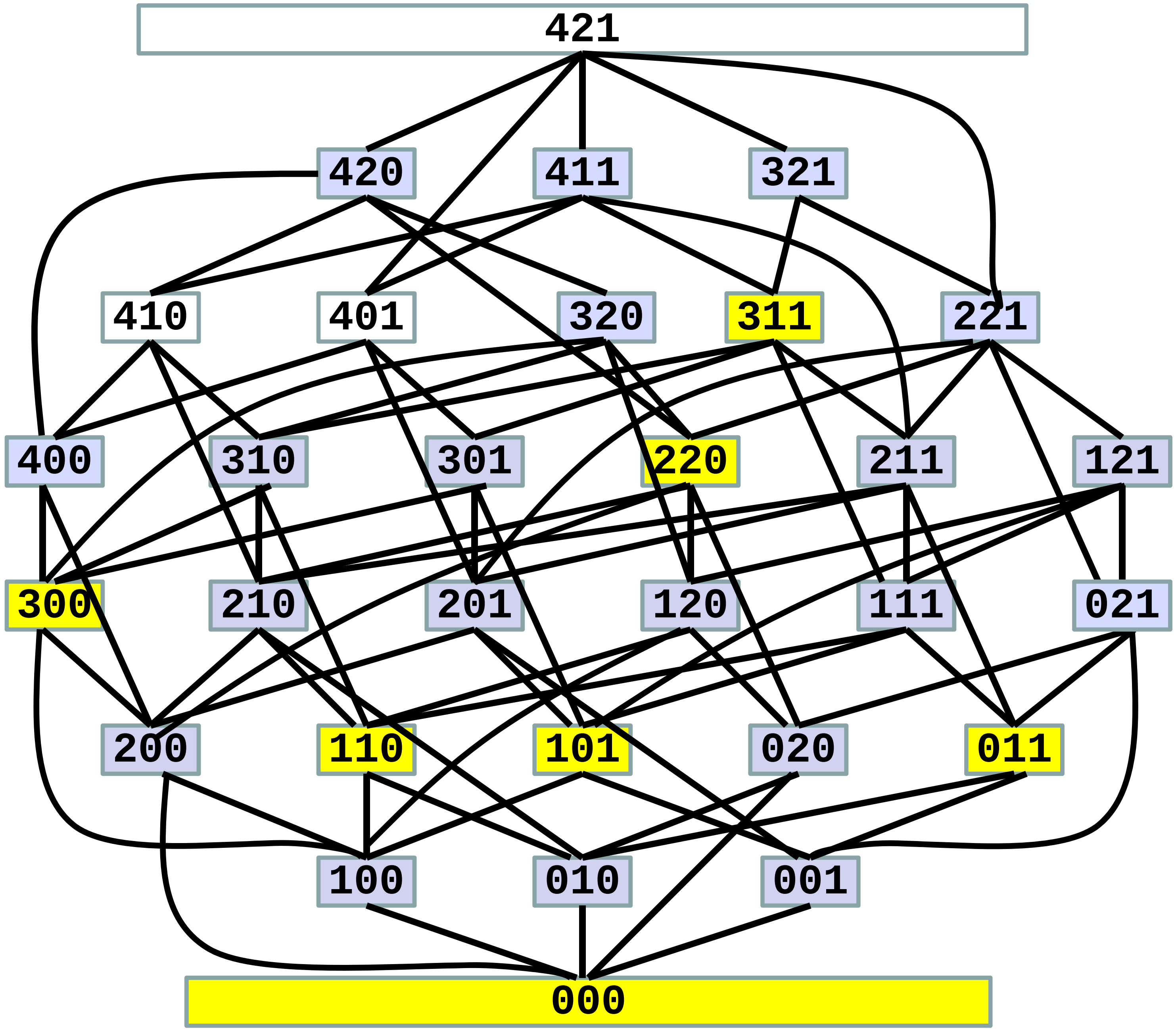
Player who removes the last peg wins.

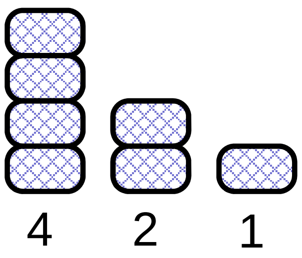




Each player can remove 1 or 2 pegs.

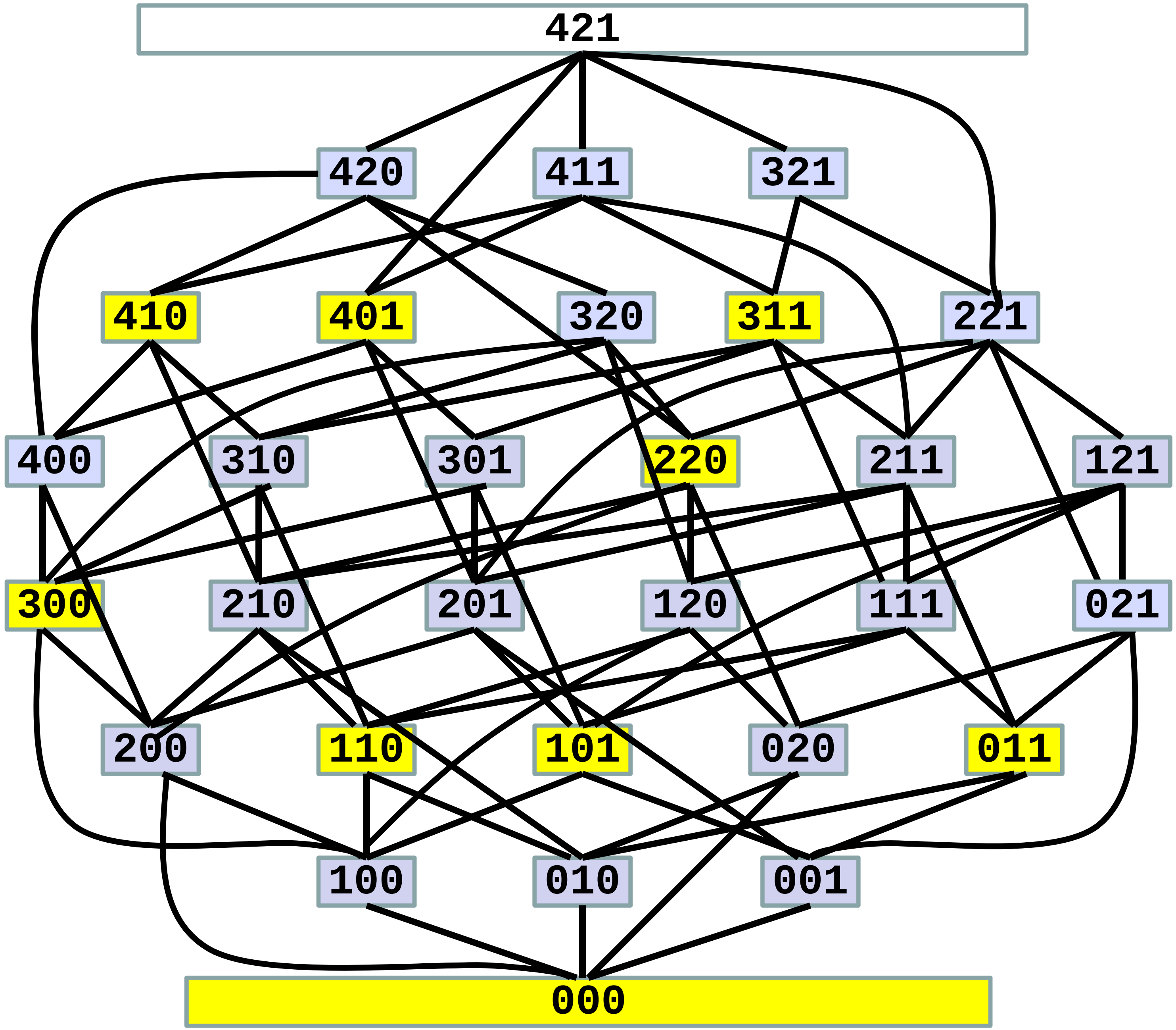
Player who removes the last peg wins.

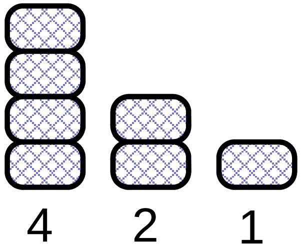




Each player can remove 1 or 2 pegs.

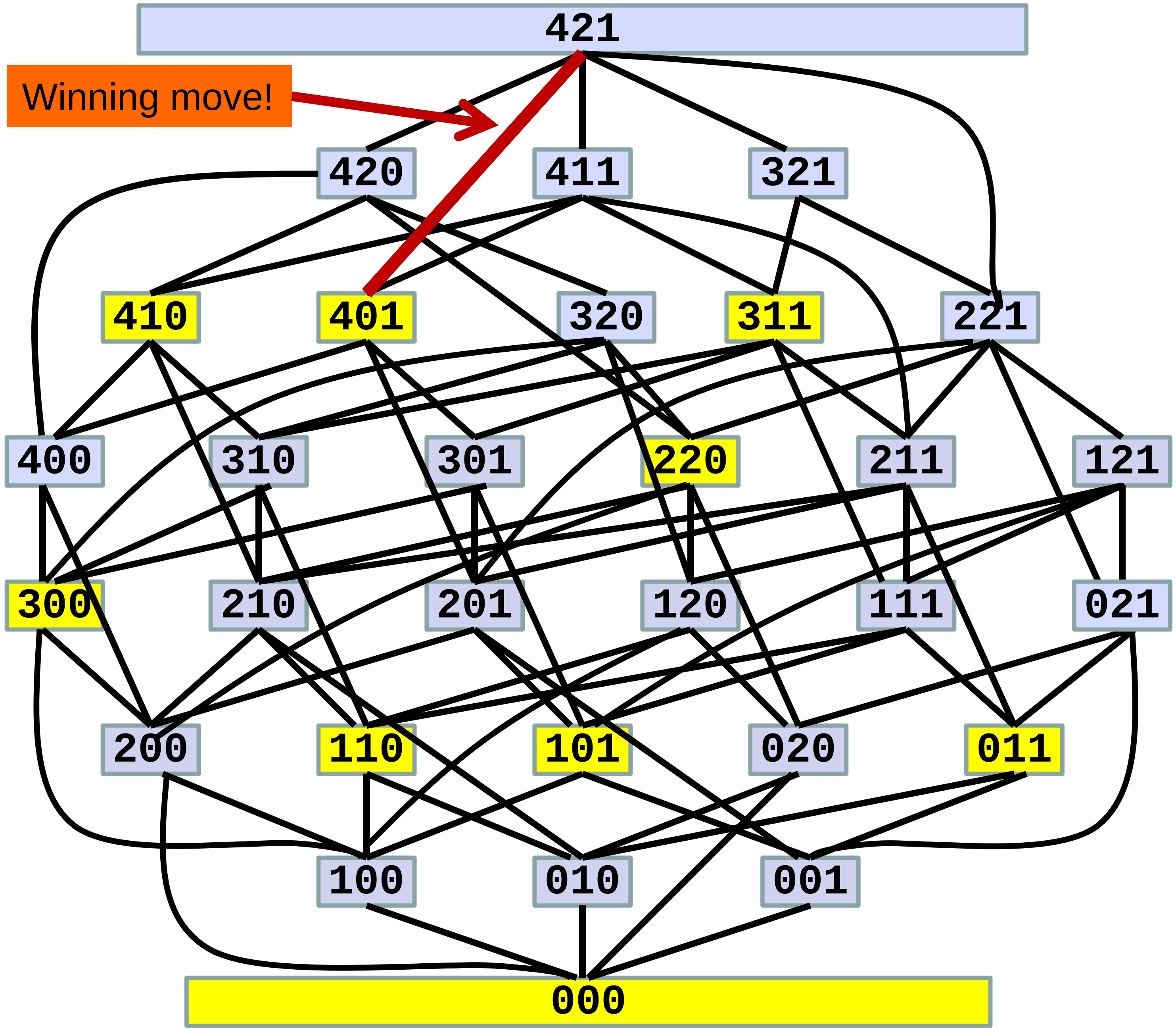
Player who removes the last peg wins.

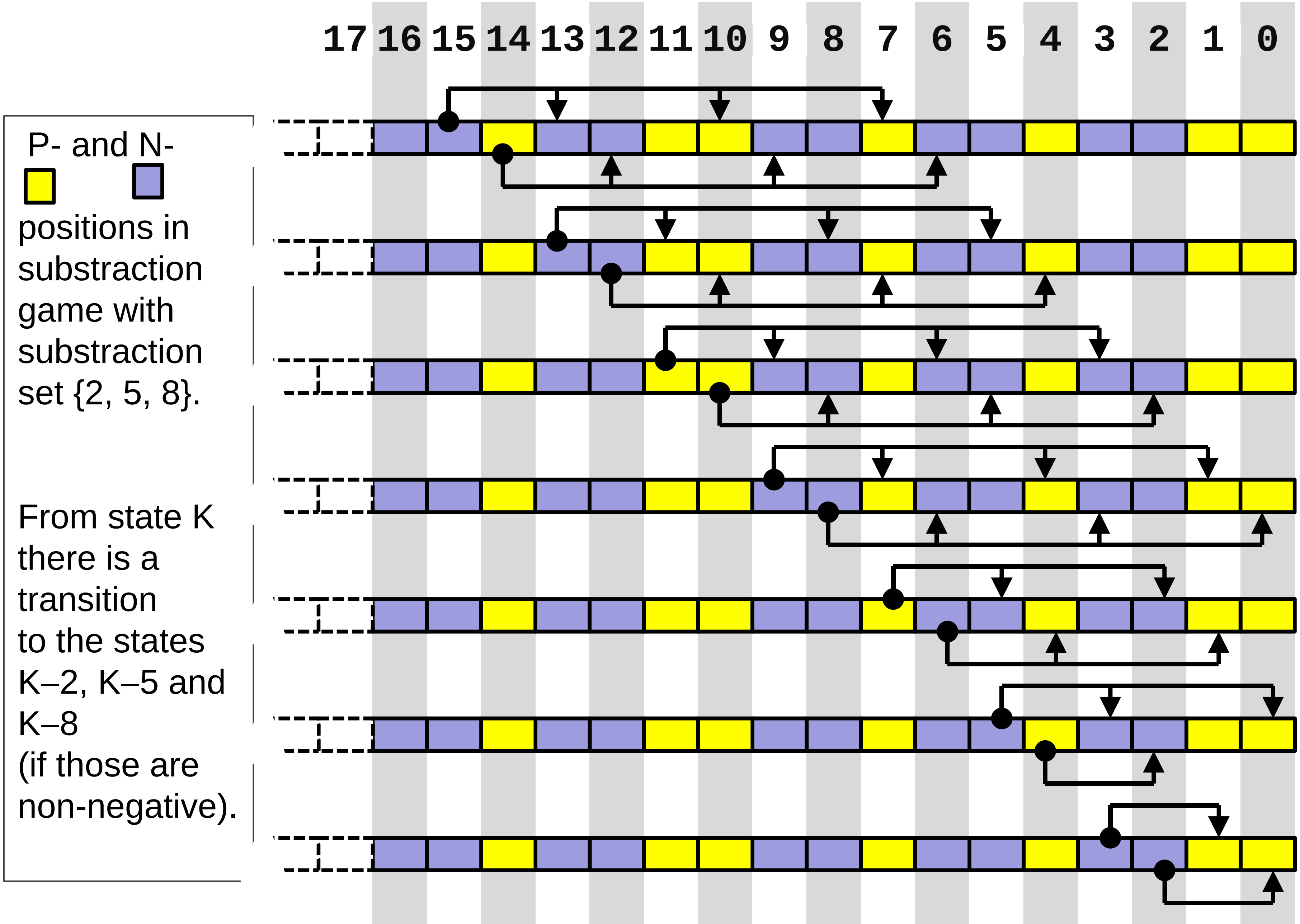




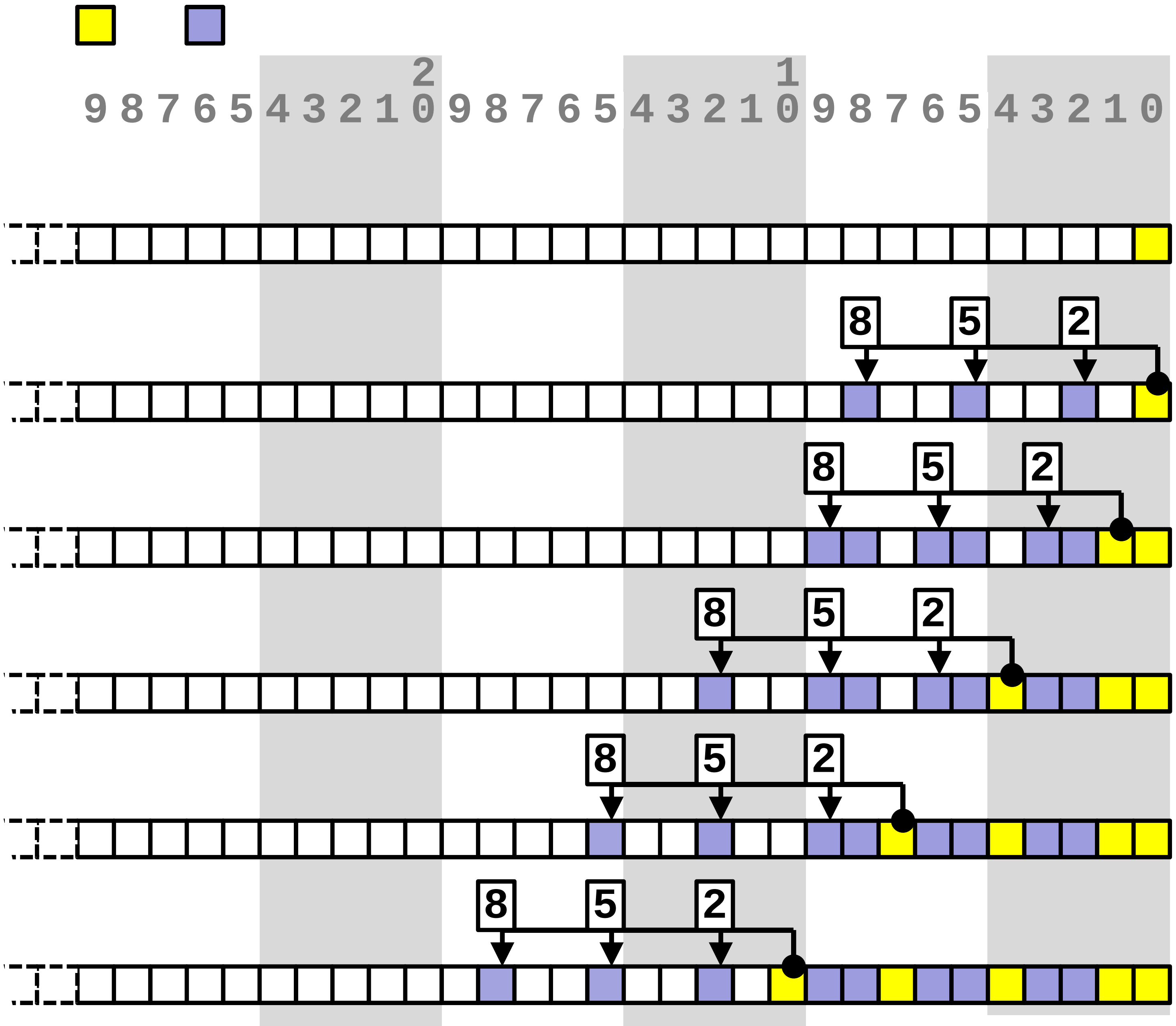
Each player can remove 1 or 2 pegs.

Player who removes the last peg wins.

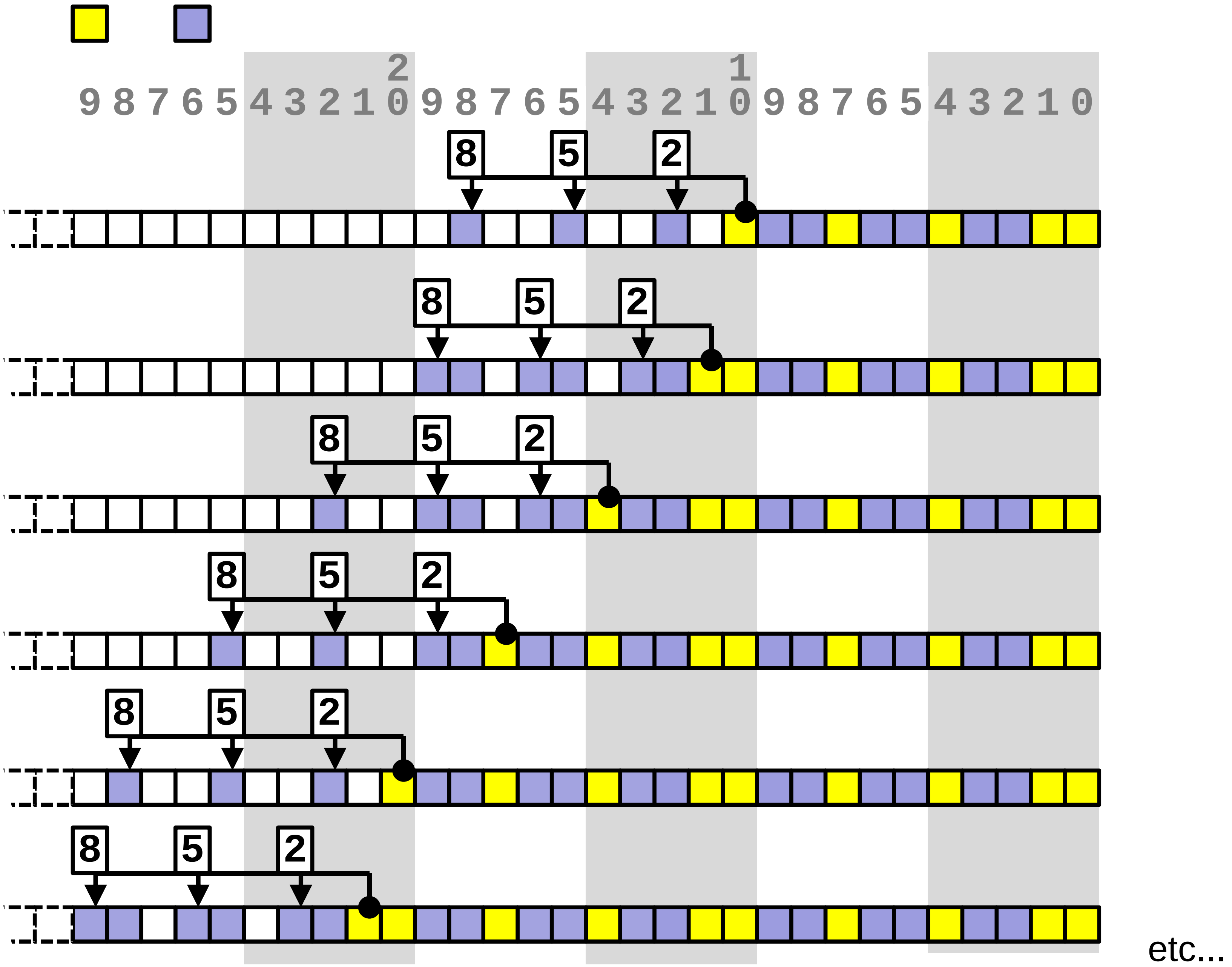



17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
P- and N-
positions in
substraction
game with
substraction
set {2, 5, 8}.
From state K
there is a
transition
to the states
K–2, K–5 and
K–8
(if those are
non-negative).

Generate P and N positions in the substraction game with substraction set {2, 5, 8}.



Generate P and N positions in the substraction game with substraction set {2, 5, 8}.



Binary representation of positive integers in Fibonacci base

N =	... 34	21	13	8	5	3	2	1
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	1	0	0
4	0	0	0	0	0	1	0	1
5	0	0	0	0	1	0	0	0
6	0	0	0	0	1	0	0	1
7	0	0	0	0	1	0	1	0
8	0	0	0	1	0	0	0	0
9	0	0	0	1	0	0	0	1
10	0	0	0	1	0	0	1	0
11	0	0	0	1	0	1	0	0
12	0	0	0	1	0	1	0	1
13	0	0	1	0	0	0	0	0
14	0	0	1	0	0	0	0	1
15	0	0	1	0	0	0	1	0
16	0	0	1	0	0	1	0	0
17	0	0	1	0	0	1	0	1
18	0	0	1	0	1	0	0	0
19	0	0	1	0	1	0	0	1
20	0	0	1	0	1	0	1	0
21	0	1	0	0	0	0	0	0
22	0	1	0	0	0	0	0	1
23	0	1	0	0	0	0	1	0
24	0	1	0	0	0	1	0	0

N =	... 55	34	21	13	8	5	3	2	1
25	0	0	1	0	0	0	1	0	1
26	0	0	1	0	0	1	0	1	0
27	0	0	1	0	0	1	1	0	1
28	0	0	1	0	0	1	0	1	0
29	0	0	1	0	1	0	0	0	0
30	0	0	1	0	1	0	0	0	1
31	0	0	1	0	1	0	0	1	0
32	0	0	1	0	1	0	1	0	0
33	0	0	1	0	1	0	1	0	1
34	0	1	0	0	0	0	0	0	0
35	0	1	0	0	0	0	0	0	1
36	0	1	0	0	0	0	0	1	0
37	0	1	0	0	0	0	1	0	0
38	0	1	0	0	0	0	1	0	1
39	0	1	0	0	0	1	0	0	0
40	0	1	0	0	0	1	0	0	1
41	0	1	0	0	0	0	0	1	0
42	0	1	0	0	1	0	0	0	0
43	0	1	0	0	1	0	0	0	1
44	0	1	0	0	1	0	0	1	0
45	0	1	0	0	1	0	1	0	0
46	0	1	0	0	1	0	1	0	1
47	0	1	0	1	0	0	0	0	0
48	0	1	0	1	0	0	0	0	1

Fibonacci Nim:

One pile of tokens.
First player can remove 1 or more tokens but not all of them.
Next, each player than can remove at most twice the number of tokens removed in his oponent's last move.
Player who removes last token wins.

Let:
N be the current number of tokens.
RemLim be maximum tokens which can be currently removed.
Fmin be the rightmost base element present in N (rightmost 1).

Then:
RemLim < **Fmin** .... P-position
RemLim >= **Fmin** ....N-position

Rule:
In N-position remove **Fmin** tokens.

base:

N =	... 55	34	21	13	8	5	3	2	1
25	0	0	1	0	0	0	1	0	1
26	0	0	1	0	0	1	0	1	0
27	0	0	1	0	0	1	1	0	1
28	0	0	1	0	0	1	0	1	0
29	0	0	1	0	1	0	0	0	0
30	0	0	1	0	1	0	0	0	1
31	0	0	1	0	1	0	0	1	0
32	0	0	1	0	1	0	1	0	0
33	0	0	1	0	1	0	1	0	1
34	0	1	0	0	0	0	0	0	0
35	0	1	0	0	0	0	0	0	1
36	0	1	0	0	0	0	0	1	0
37	0	1	0	0	0	0	1	0	0
38	0	1	0	0	0	0	1	0	1
39	0	1	0	0	0	1	0	0	0
40	0	1	0	0	0	1	0	0	1
41	0	1	0	0	0	0	0	1	0
42	0	1	0	0	1	0	0	0	0
43	0	1	0	0	1	0	0	0	1
44	0	1	0	0	1	0	0	1	0
45	0	1	0	0	1	0	1	0	0
46	0	1	0	0	1	0	1	0	1
47	0	1	0	1	0	0	0	0	0
48	0	1	0	1	0	0	0	0	1

RemLim < Fmin .... P-position
RemLim >= Fmin ....N-position

Example:
Pile with 45 tokens.

First move:
N = 45, RemLim = 44, Fmin = 3.
RemLim >= Fmin .... N-position
Remove Fmin: N = 45 –3 = 42

Next move:
The opponent can remove 1 to 6 tokens, that is, he can set the pile to 41, 40, 39, 38, 35, 35 tokens.

All these are N-positions, because RemLim = 6, Fmin <= 5.

base:

N =	... 55	34	21	13	8	5	3	2	1
25	0	0	1	0	0	0	1	0	1
26	0	0	1	0	0	1	0	1	0
27	0	0	1	0	0	1	1	0	1
28	0	0	1	0	0	1	0	1	0
29	0	0	1	0	1	0	0	0	0
30	0	0	1	0	1	0	0	0	1
31	0	0	1	0	1	0	0	1	0
32	0	0	1	0	1	0	1	0	0
33	0	0	1	0	1	0	1	0	1
34	0	1	0	0	0	0	0	0	0
35	0	1	0	0	0	0	0	0	1
36	0	1	0	0	0	0	0	1	0
37	0	1	0	0	0	0	1	0	0
38	0	1	0	0	0	0	1	0	1
39	0	1	0	0	0	1	0	0	0
40	0	1	0	0	0	1	0	0	1
41	0	1	0	0	0	0	0	1	0
42	0	1	0	0	1	0	0	0	0
43	0	1	0	0	1	0	0	0	1
44	0	1	0	0	1	0	0	1	0
45	0	1	0	0	1	0	1	0	0
46	0	1	0	0	1	0	1	0	1
47	0	1	0	1	0	0	0	0	0
48	0	1	0	1	0	0	0	0	1

RemLim < Fmin .... P-position
RemLim >= Fmin ....N-position

Example continues:
Opponent took 4.
Pile with 38 tokens.

Next move:
N = 38, RemLim = 8, Fmin = 1.
RemLim >= Fmin .... N-position
Remove Fmin: N = 38 –1 = 37

Next move:
The opponent can remove 1 or 2 tokens, that is, he can set the pile to 36 or 35 tokens.

All these are N-positions, because RemLim = 2, Fmin <= 3.

base:

N = / ...	55	34	21	13	8	5	3	2	1
25	0	0	1	0	0	0	1	0	1
26	0	0	1	0	0	1	0	1	0
27	0	0	1	0	0	1	1	0	1
28	0	0	1	0	0	1	0	1	0
29	0	0	1	0	1	0	0	0	0
30	0	0	1	0	1	0	0	0	1
31	0	0	1	0	1	0	0	1	0
32	0	0	1	0	1	0	1	0	0
33	0	0	1	0	1	0	1	0	1
34	0	1	0	0	0	0	0	0	0
35	0	1	0	0	0	0	0	0	1
36	0	1	0	0	0	0	0	1	0
37	0	1	0	0	0	0	1	0	0
38	0	1	0	0	0	0	1	0	1
39	0	1	0	0	0	1	0	0	0
40	0	1	0	0	0	1	0	0	1
41	0	1	0	0	0	0	0	1	0
42	0	1	0	0	1	0	0	0	0
43	0	1	0	0	1	0	0	0	1
44	0	1	0	0	1	0	0	1	0
45	0	1	0	0	1	0	1	0	0
46	0	1	0	0	1	0	1	0	1
47	0	1	0	1	0	0	0	0	0
48	0	1	0	1	0	0	0	0	1