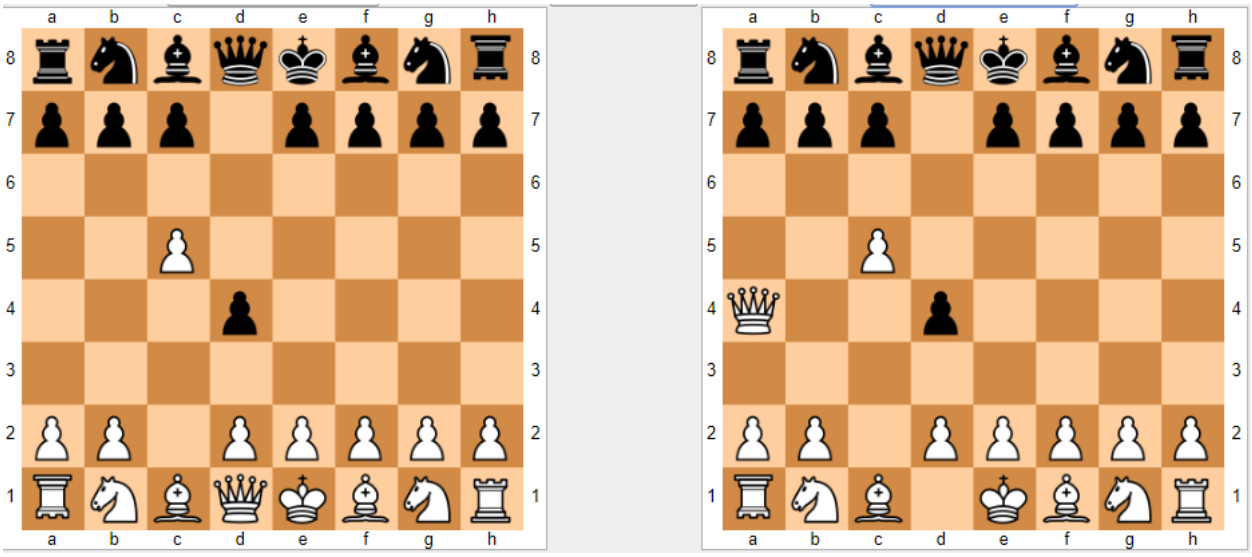
CSE 415, Univ. of Wash. Name: _____

Zobrist Hashing (SOLUTIONS)

An important property of the hash functions used in Zobrist hashing is that they can be computed quickly. Also, collisions tend to be unlikely when the technique is applied properly. Two distinct states of a game of Baroque Chess are shown below. Suppose that the board on the left shows the "current state" s, and the board on the right shows s' one of its successors.



Current state s successor state s'

Suppose that the hash code for the current state is h(s) = 00100110.

Suppose that the precomputed random number for the white Withdrawer at square d1 is WWD1 = 01001101.

Suppose that the precomputed random number for the white Withdrawer at square a4 is WWA4 = 11010111.

Suppose Zobrist hashing is in use, and a portion of the hash table is shown below. In this simple case, each hash-code column entry is simply the array index (given in binary) for one row of the hash table.

Then determine what information, if any, has already been stored information for state s'.

Hash code	State	n-ply used	value
10111001	<state object ….>	3	6
10111010		undefined	undefined
10111011	<state object ….>	2	-7
10111100	<state object ….>	2	4
10111101		undefined	undefined
10111110	<state object ….>	2	-10

h(s): 00100110
WWD1: 01001101
h(s) XOR WWD1: 01101011
WWA4: 11010111
h(s) XOR WWD1 XOR WWA4:
h(s'): 10111100

n-ply used: 2
value: 4

To do this, you should take h(s') = h(s) XOR WWD1 XOR WWA4. Then retrieve the corresponding n-ply used and value.

Here we are using 8-bit codes, though in practice 24 or more bits should be used. Commonly, 64-bit hash codes are used, and then the actually occurring hash codes are stored in a table instead of the states themselves. The possibility of collisions then might be ignored, to save time, at a slight risk of getting inaccurate results during play.