

Tutorial 04 Zobrist Keys

Not to be redistributed
to Course Hero or any
other public websites

Introduction

In this tutorial, you will gain better understanding about Zobrist keys and how they can be used for hashing game states. Refer to slides 28-34 of Lecture 02 handout for completing this tutorial.

Task 1. Initializing Zobrist Keys

The Jungle Chess has the following game board and chess pieces:

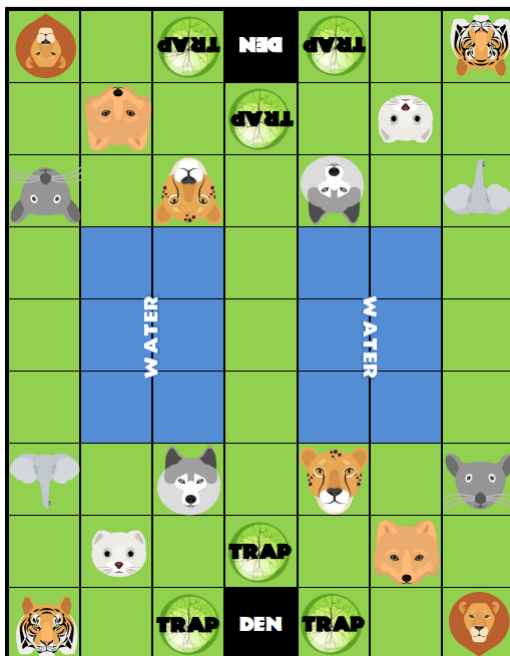


Image source: <https://owlworkslc.com/featured-game/its-a-jungle-out-there/>

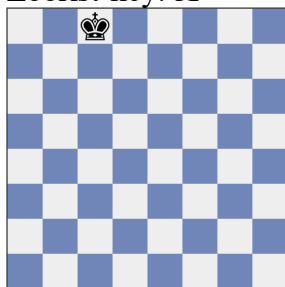
Each of the 2 players has 8 different chess pieces representing different animals: 1. Rat; 2. Cat; 3. Dog; 4. Wolf; 5. Leopard; 6. Tiger; 7. Lion; 8. Elephant. The game board can be visualized as a 9×7 square board. Assuming that each square on the Jungle chess game board can be occupied by any of the animal chess pieces, how many random Zobrist keys need to be initialized if Zobrist hashing is used to implement the transposition table for representing the board states of this game?

Note that you do not need to know the rules of this game in order to answer this question.

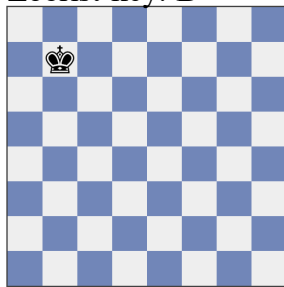
Task 2. Incremental Zobrist Hashing

The Zobrist keys for some of the chess game board states are represented by the variables A-J shown in the following figure:

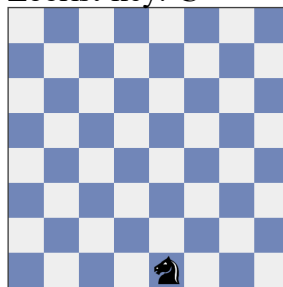
Zobrist key: A



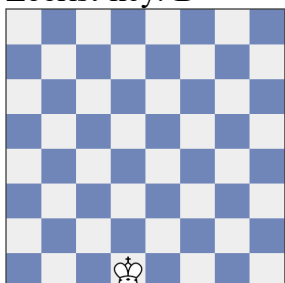
Zobrist key: B



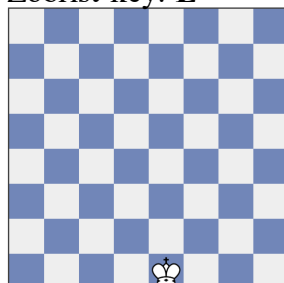
Zobrist key: C



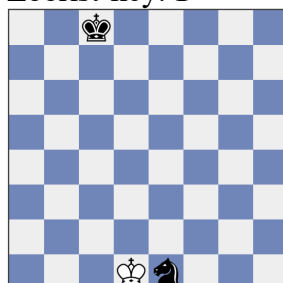
Zobrist key: D



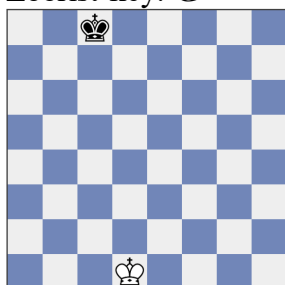
Zobrist key: E



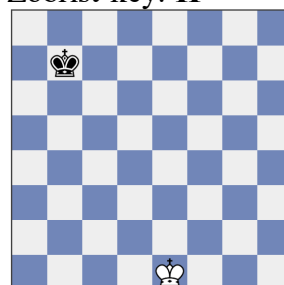
Zobrist key: F



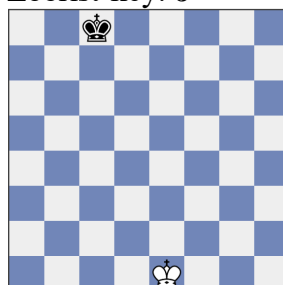
Zobrist key: G



Zobrist key: H



Zobrist key: J



How can the Zobrist key J be derived from the other Zobrist keys A-H? Is there only one way to obtain J from A-H?

Task 3. Complete the Canvas Quiz

Complete the quiz "Tutorial 04" on the [Canvas](#) course page (Assignments > Tutorial 04 before the end of the tutorial in the CS lab.

Task 4. Implement the Zobrist Hashing for Tic-Tac-Toe

Implement the Zobrist Hasing for 3×3 Tic-Tac-Toe (refer to slide 29 on Lecture 02).

- Use 8 bits as the length of each Zobrist key. From your work on Tutorial 02, you should have the code that iterates through all the possible game board states and identifies those boards that can be generated from valid games. Now add the code to determine the Zobrist key for each board that can be generated from a valid game. Further note down if a clash (collision) occurs and count the number of times for clashing. Do you expect clashing would definitely occur with this set of 8-bit Zobrist keys?
- Now use 16 bits as the length of each Zobrist key and repeat your work as in a). Do you expect clashing would occur more often or less often than in a)?