

Re: Chess boards & connections.

Re: Chess boards & connections.

Source: <http://sci.tech--archive.net/Archive/sci.math/2006-04/msg03975.html>

- *From:* dynamics@xxxxxxxxxxxxx
 - *Date:* 20 Apr 2006 23:21:35 -0700
-

Toshi ikutsu? wrote:

<dynamics@xxxxxxxxxxxxx> wrote in message
news:1145569228.272184.92380@xx

Michael Stemper wrote:

In article
<114555339.841442.203250@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
dynamics writes:

Trying to calculate if I can write a Chess AI.

I need to define all possible boards.

I have a total of 64 different pieces, 16 for
Black
and 16 for White to start, and since each
pawn
can be promoted, to either a Queen or
Knight, a

They can't be promoted to a bishop or a rook?

LOL, then you'll totally fuck my math, actually
I have a patch, but you're right ;-)

what about "in passing" for pawns?
and castling?

further 16 for B&W's 8 pawns for another

Re: Chess boards & connections.

32.

There are 65 locations on the board, $8*8 + 1$ for non-existence called the side bar.

So I dimension an Array, (64,65) where the 64 provides the *serial number* for all possible pieces and each of those can be in 65 locations, providing $64*65=4160$ boards.

Nope.

Place the first of the 64 pieces. How many choices for a square do you have? 65. Now, for each of those possibilities, you have 64 choices for placing the second piece. So, two pieces already uses up your 4160 boards. If you place a third piece, you'll have 63 available squares for each of those 4160 possibilities, or $65*64*63 = 262080$. To place 64 pieces on 65 squares — even with disallowing two pieces on the same square from the start — you're going to have 65! possible boards.

I'd say that, since each player can only have 16 pieces at any time, you number the pieces 1–16 (or 0–15 if you prefer), and keep track of what type of piece (Q, K, Kt, p, etc.) each one is. This way, you only have 32 pieces to deal with. Then, you probably should get rid of your "sidebar" and just not necessarily have all of the pieces included in the board at any time.

You are too complicated. All I needed was the 2D Array I defined by matrix (64,65) defining all boards.

too limited. try checkers instead.

Re: Chess boards & connections.

Ok, you might be right, I was thinking we could go through all 64 Pieces (as defined in the OP) and output a Location (0-64) like,

Location (P) = L

where P => 1 to 64 and L=>0 to 64,

to describe any possible chess board.

Is that true?

This strategy will reduce the number of configurations to $(64!)/(32!)$ time some multiplier based upon how many duplicate pieces a player has.

(I'll skip using exclusion right now, like two pieces can't occupy the same square, but all that means is there are < 4160 possible boards).

Nope. Like I pointed out above, even if you build in "exclusion" from the start, you'll need to add a lot of zeros to the right of "4160"

Well then provide the needed Array, I'll repeat I get a (64,65) for ALL configurations, including many that are impossible.

if your math skills indicate your programming skills, you are a starting hack-er.

Yup, trying to learn algebra.

Ken

.