```
%Facts
bd(X11,X12,X13,X21,X22,X23,X31,X32,X33).

%Possible Moves
move(
    _,
    bd(b,X12,X13,X21,X22,X23,X31,X32,X33),
    bd(X12,b,X13,X21,X22,X23,X31,X32,X33)
).
move(
    _,
    bd(b,X12,X13,X21,X22,X23,X31,X32,X33),
    bd(X21,X12,X13,b,X22,X23,X31,X32,X33)
).
move(
    _,
    bd(X11,X12,X13,b,X22,X23,X31,X32,X33),
    bd(b,X12,X13,X11,X22,X23,X31,X32,X33)
).
move(
    _,
    bd(X11,X12,X13,b,X22,X23,X31,X32,X33),
    bd(X11,X12,X13,X31,X22,X23,b,X32,X33)
).
move(
    _,
    bd(X11,X12,X13,b,X22,X23,X31,X32,X33),
    bd(X11,X12,X13,X22,b,X23,X31,X32,X33)
).
move(
    _,
    bd(X11,X12,X13,X21,b,X23,X31,X32,X33),
    bd(X11,X12,X13,b,X21,X23,X31,X32,X33)
).
move(
    N,
    bd(X11,X12,X13,X21,X22,X23,X31,X32,X33),
    bd(Y31,Y21,Y11,Y32,Y22,Y12,Y33,Y23,Y13)
):-
    N<4,
    N1 is N + 1,
    move(
        N1,
        bd(X13,X23,X33,X12,X22,X32,X11,X21,X31),
        bd(Y11,Y12,Y13,Y21,Y22,Y23,Y31,Y32,Y33)
    ).

%Helpers

printBoards([]).
printBoards([B|Bs]):-printBoard(B), printBoards(Bs).

printBoard(bd(X11,X12,X13,X21,X22,X23,X31,X32,X33)):-
    format(
        "~n|~t~a~t~4||~t~a~t~4+|~t~a~t~4+|~n",
        [X11,X12,X13]
    ),
    format(
        "|~t~a~t~4||~t~a~t~4+|~t~a~t~4+|~n",
        [X21,X22,X23]
    ),
    format(
        "|~t~a~t~4||~t~a~t~4+|~t~a~t~4+|~n",
        [X31,X32,X33]
    ).

quicksort([],[]).
quicksort([X|Xs],Ys) :-
    partition(Xs,X,Left,Right),
    quicksort(Left,Ls),
    quicksort(Right,Rs),
    append(Ls,[X|Rs],Ys).
```

```
partition([], _, [], []).
partition([[K,Ks,H1]|L], [X,Xs,H2], M, [[K,Ks,H1]|N]):-
    X > K, !,
    partition(L, [X,Xs,H2], M, N).
partition([K|L], X, [K|M], N):-
    partition(L, X, M, N).

append([],Ys,Ys).
append([X|Xs],Ys,[X|Zs]) :- append(Xs,Ys,Zs).

%GameSolver
createValuedBoardList([],Y, Y).
createValuedBoardList([X|XS],Y, R):-
    evaluate(X, V),
    createValuedBoardList(XS, [[V,X]|Y], R).

evaluate(
    bd(X11,X12,X13,X21,X22,X23,X31,X32,X33),
    R):-
        eHelper([X11,X12,X13,X21,X22,X23,X31,X32,X33],0,1,R).

eHelper(_,V,10,V).
eHelper([F|Fs],V,N,R):-
    F=N,
    V1 is V+1,
    N1 is N+1,
    eHelper(Fs, V1, N1, R).
eHelper([F|Fs],V,N,R):-
    N1 is N+1,
    eHelper(Fs, V, N1, R).

solveGame(BG):-
    createValuedBoardList([BG],[], ValuedStart),
    addPath(ValuedStart, [],[], R),
    solver(R,[BG]).

addPath([], _,Y, Y).
addPath([[V, BD]|Xs], H,Y, R):- addPath(Xs, H, [[V, BD,H]|Y], R).

solver([[8,bd(1,2,3,4,5,6,7,8,b), Path]|VBDs], _):-
    append(Path, [bd(1,2,3,4,5,6,7,8,b)], FinalPath),
    printBoards(FinalPath).
solver([[Value, BD, Path]|VBDs], KnownBD):-
    findall(Y,(move(1,BD,Y), not(member(Y, KnownBD))), Moves),
    append(Moves, KnownBD, KnownBDNew),
    createValuedBoardList(Moves,[], ValuedMoves),
    append(Path, [BD], NewPath),
    addPath(ValuedMoves, NewPath,[], ValuedMovesPath),
    append(VBDs, ValuedMovesPath, VBDsNew),
    quicksort(VBDsNew, SortedValuedMoves),
    solver(SortedValuedMoves, KnownBDNew).
```