```
function [stateMatrix] = successors(inpState)
%This function produces all possible successor states given the input
%state.
%Each column of the output represents one of the successor nodes.

parentState = inpState;
num = length(parentState); % Total number of elements in a puzzle
 state
lengthEdge = sqrt(num);

[~, indGap] = min(parentState); % Detect location of the gap (Note
 that
...the gap is represented by a "0" which is the minimum element of the
 state array.)

rowGap = ceil(indGap/ lengthEdge); % Column number of the gap
colGap = indGap - lengthEdge * (rowGap-1); % Row number of the gap

stateMatrix = []; % Initialize the state matrix

if (rowGap ~= lengthEdge)
    % The gap can be shifted through downwards
    dummyState = parentState;
    dummyState(indGap) = dummyState(indGap+lengthEdge);
    dummyState(indGap+lengthEdge) = 0;
    stateMatrix = [stateMatrix dummyState];
end

if (colGap ~= 1)
    % The gap can be shifted through left
    dummyState = parentState;
    dummyState(indGap) = dummyState(indGap-1);
    dummyState(indGap-1) = 0;
    stateMatrix = [stateMatrix dummyState];
end

if (colGap ~= lengthEdge)
    % The gap can be shifted through right
    dummyState = parentState;
    dummyState(indGap) = dummyState(indGap+1);
    dummyState(indGap+1) = 0;
    stateMatrix = [stateMatrix dummyState];
end

if (rowGap ~= 1)
    % The gap can be shifted through upwards
    dummyState = parentState;
    dummyState(indGap) = dummyState(indGap-lengthEdge);
    dummyState(indGap-lengthEdge) = 0;
    stateMatrix = [stateMatrix dummyState];
end
```

```
end


% function [nodeID] = convert_state_to_NodeID(inpState)
%     nodeID = num2str(inpState);
%     nodeID = str2num(nodeID(~isspace(nodeID)));
% end
```

Published with MATLAB® R2017b