8 puzzle problem using iterative depth search bfs.

```
def bfs ( src, target):

        visited_states = [ ]
        visited_states . append (src)
        arr =  [src]
        c = 0
        while arr :
                  c += 1
                  if arr[0] == target:
                          return True
                  arr += possible_moves(arr[0], visited_stat
                  arr. pop(0)
          return false.

def    possible_moves( state, visited_states):

        b = state. index(-1)
        d = [ ]

        if  b+3  in  range(9):
                  d. append ('d')
        if  b-3  in  range (9):
                  d. append ('u')
         if b   not in [0, 3, 6]:
                  d. append ('l')
          if b   not in [2, 5, 8]:
                   d. append ('r')

        pos_moves  = [ ]
```

```
    for move in d:
            pos-moves.append(gen( state, move, b))

    return [ move for move in pos-moves if move
          not in visited_states]

def gen ( state, direction, blank_spot ):
      temp = state.copy()

      if direction == 'd':
          a= temp[ blank_spot + 3]
          temp [ blank. spot + 3] = temp[blank . spot]
          temp[ blank-spot] = a

      elif direction == 'u':
           a = temp[ blank. spot - 3]
           temp[ blank. spot - 3] = temp[ blank. spot]
           temp[ blank- spot] = a

      elif direction == 'l':
            a = temp[ blank. spot -1]
           temp [ blank spot -1] = temp[ blan - spot]
            temp[ blan - spot] = a

      elif direction == 'r':
            a = temp[ blan - spot +1]
           temp[ ~~bla~~ + 1] = temp[ blank_spot]
           temp [ blan_ spot] = a

    return temp.
```

Scanned with CamScanner