DFS

Função de Direções

- Função Play -> direções do quebra-cabeças
- Setando estados iniciais e finais

In [9]:
```
import copy
import time
```

In [10]:
```
# Caso Base => 0 1 2 3 4 5 6 7 8
initialState1 = [[-1,1,2], [7,8,3], [6,5,4]]

# Case 2 => 1,5,4, 3,7,2, 6,8,0
#initialState1 = [[1,5,4],[3,7,2],[6,8,-1]]
finalState1 = [[1,2,3], [8,-1,4], [7,6,5]]
t0 = time.time()

def play(puzzle, direction):
  if direction == "cima":
    for i in range(3):
      for j in range(3):
        if(puzzle[i][j]==-1):
          if i!=0:
            puzzle[i][j] = puzzle[i-1][j]
            puzzle[i-1][j] = -1
          return puzzle

  if direction=="baixo":
    for i in range(3):
      for j in range(3):
        if(puzzle[i][j]==-1):
          if i!=2:
            puzzle[i][j] = puzzle[i+1][j]
            puzzle[i+1][j] = -1
          return puzzle

  if direction=="esquerda":
    for i in range(3):
      for j in range(3):
        if(puzzle[i][j] == -1):
          if j!=0:
            puzzle[i][j] = puzzle[i][j-1]
            puzzle[i][j-1] = -1
          return puzzle

  if direction == "direita":
    for i in range(3):
      for j in range(3):
        if(puzzle[i][j] == -1):
          if j!=2:
            puzzle[i][j] = puzzle[i][j+1]
            puzzle[i][j+1] = -1
          return puzzle
```

Função DFS

In [11]:
```
def buscaProfundidade():
  items = []
  passos = 0
  initialFormatted = [initialState1,"start"]
  items.append(initialFormatted)

  while(True):
    passos += 1
    puzzle = items.pop(0)
    print(puzzle[1])

    if(puzzle[0] == finalState1):
      print('Encontrado após: '+str(passos - 1) + ' Iterações')
      break
    else:
      if(puzzle[1]!="baixo"):
        puzzleCopia = copy.deepcopy(puzzle[0])
        movimento = play(puzzleCopia, "cima")

        if(movimento!=puzzle[0]):
          items.insert(0, [movimento,"cima"])

      if(puzzle[1]!="direita"):
        puzzleCopia = copy.deepcopy(puzzle[0])
        movimento = play(puzzleCopia, "esquerda")

        if(movimento != puzzle[0]):
          items.insert(0, [movimento,"esquerda"])

      if(puzzle[1]!="cima"):
        puzzleCopia = copy.deepcopy(puzzle[0])
        movimento = play(puzzleCopia, "baixo")

        if(movimento != puzzle[0]):
          items.insert(0, [movimento,"baixo"])

      if(puzzle[1]!="esquerda"):
        puzzleCopia = copy.deepcopy(puzzle[0])
        movimento = play(puzzleCopia, "direita")

        if(movimento != puzzle[0]):
          items.insert(0, [movimento,"direita"])
buscaProfundidade()
t1 = time.time()
print('Tempo decorrido:', t1-t0)
print('------')
```

```
start
direita
direita
baixo
baixo
esquerda
esquerda
cima
direita
Encontrado após: 8 Iterações
Tempo decorrido: 0.5355398654937744
------
```

In [ ]:

In [ ]: