**Trabalho 2 - Análise de Algoritmos**

**Alunos:** Lucas Braz e Tatiana Reimer Barata

**Tarefa 1: Criação do grafo de espaço de estados**

Criamos uma lista, chamada *states* para guardar todos os estados possíveis, que são representados por matrizes, e um dicionário *graph* para guardar os estados vizinhos desses estados. O zero indica o espaço vazio no puzzle.

Por exemplo:

Diagrama, Desenho técnico

Descrição gerada automaticamente

A lista *states* desse grafo será:

Número do estado - Estado

0 - [[1, 2, 3], [0, 4, 6], [7, 5, 8]]

1 - [[0, 2, 3], [1, 4, 6], [7, 5, 8]]

2 - [[1, 2, 3], [4, 0, 6], [7, 5, 8]]

3 - [[1, 2, 3], [7, 4, 6], [0, 5, 8]]

4 - [[2, 0, 3], [1, 4, 6], [7, 5, 8]]

E o dicionário *graph* será:

graph = { 0: [1, 2, 3], 1: [0, 4], 2: [0], 3: [0], 4: [1] }

Então, por exemplo, o estado 0 pode passar para os estados 1,2 e 3.

**Respostas:**

**1.** O número total de configurações é 9! / 2 = 181440

**2.** Quando é possível passar de um estado para outro em um único movimento esses nós são conectados por uma aresta

**3.** Dois nós que tem o mesmo nó pai não têm uma aresta entre eles

**Tarefa 2: Implementação da BFS e contagem de componentes conexos**

**Respostas:**

**1.**

Texto

Descrição gerada automaticamente

**2.** O número de componentes conexos é 1 porque o grafo é uma árvore

**Tarefa 3: Caminho mais curto**

**Respostas:**

**1.**

**2.**