

Algoritmos de Busca{

[Problema Régua-Puzzle]

Ana Kapps, André Reis e Felipe Vidal

}

```
1  Tabela de 'Conteúdo' {
2
3
4      01  Régua-Puzzle
5          Descrição do Problema
6
7          02  Metodologia e Organização
8              Linguagem, Estrutura e
9              Arquitetura
10
11              03  Execução e Avaliação
12                  Estatísticas, Análise e
13                  Dificuldades
14  }
```

1
2 01 {
3
4

5 [Régua-Puzzle]
6
7

8 Descrição do problema
9
10

11 }
12
13
14

```
1  Jogo {
2
3      |
4      |   - 2N blocos alinhados em uma régua
5      |   - 2N + 1 posições
6      |   - N blocos pretos (B)
7      |   - N blocos brancos (W)
8      |   - 1 posição vazia (-)
9
10     |
11     |   

|   |   |   |   |   |
|---|---|---|---|---|
| W | B | - | B | W |
|---|---|---|---|---|


20
21     |
22 }
```

```
1  Objetivo; {
2
3
4      - Brancos do lado esquerdo e pretos do lado
5        direito
6
7
8      - A solução ótima é aquela com menor número
9        de movimentos de menor custo
10
11
12
13 }
14
```

Movimentação {

- Bloco pula para posição vazia quando estiver distante de no máximo N casas da posição do bloco
- Custo do pulo: distância entre a posição do bloco e a posição vazia

}

ExemploDeEstadosMeta {

W	B	-	B	W
---	---	---	---	---

Estado inicial

Estados Meta:

W	W	B	-	B
---	---	---	---	---

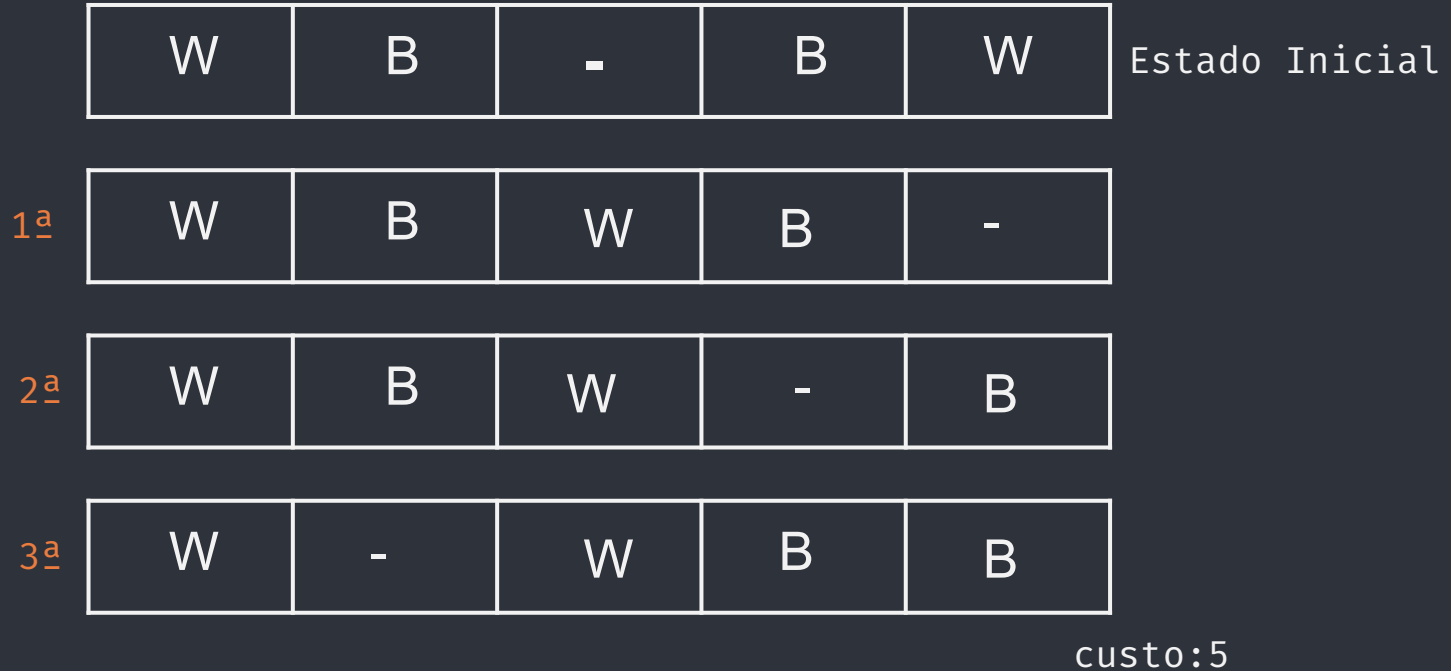
W	W	-	B	B
---	---	---	---	---

W	-	W	B	B
---	---	---	---	---

W	W	B	B	-
---	---	---	---	---

}

ExemploDeExecução {



AlgoritmosImplementados{

```
Algoritmos = [  
    'Backtracking',  
    'Busca em Largura',  
    'Busca em Profundidade',  
    'Busca Ordenada',  
    'Busca Gulosa',  
    'Busca A*',  
    'Busca IDA*',
```

```
]
```

```
}
```

1
2 02 {
3
4

5 [Metodologia e Organização]
6
7

8 Linguagem, estrutura e
9 Arquitetura
10
11

12 }
13
14

```
1 FerramentasUtilizadas {
2
3     linguagem = 'Python'
4
5
6     editores = [
7         'VS Code',
8         'PyCharm'
9     ]
10
11
12
13
14 }
```

```
1 Metodologias {
2
3
4     reuniões = 'Google Meet'
5
6
7     organizacao = 'Google Calendar'
8
9
10    gerenciamento = 'Trello'
11
12
13
14 }
```

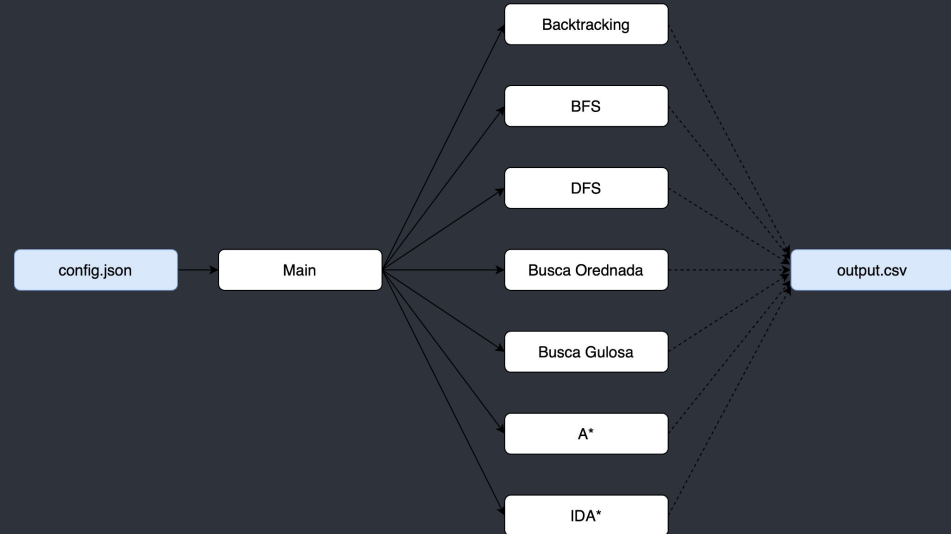
begin.py

trab.py

Arquitetura {

```
image = Image.open('architecture')
```

```
}
```



```
1  config {
2
3      "boards": [
4          {
5              "pieces": 2,
6              "initial_state": "-BWBW" // optional
7          },
8          {
9              "pieces": 3
10         }
11     ]
12 }
13
14
```

```
1  config {
2
3      // quando não é passado nenhum valor para
4      // o 'initial_state' no arquivo de
5      // configuração, ele será gerado
6      // automaticamente, seguindo no seguinte
7      // formato
8
9
10     

|   |   |   |   |   |
|---|---|---|---|---|
| W | W | - | B | B |
|---|---|---|---|---|


11
12     

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| W | W | W | - | B | B | B |
|---|---|---|---|---|---|---|


13
14 }
```

1
2 03 {
3
4

5 [Execução e Avaliação]
6
7

8 Estatísticas, Análise e
9 Dificuldades
10

11 }
12
13
14

Tempo (s)

1
2
3
4
5
6
7
8
9
10
11
12
13
14

6

5

4

3

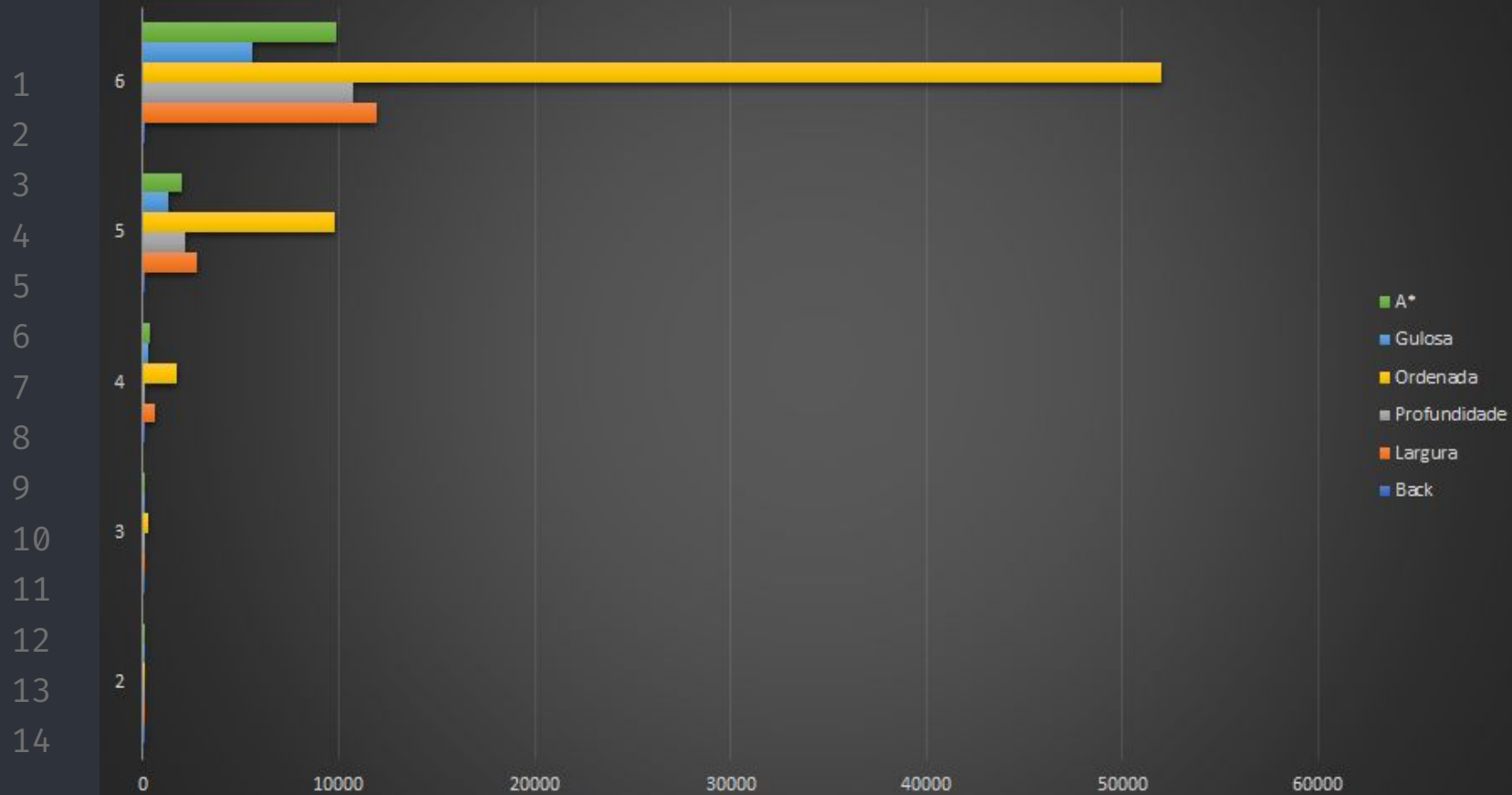
2

A*
Gulosa
Ordenada
Profundidade
Largura
Back

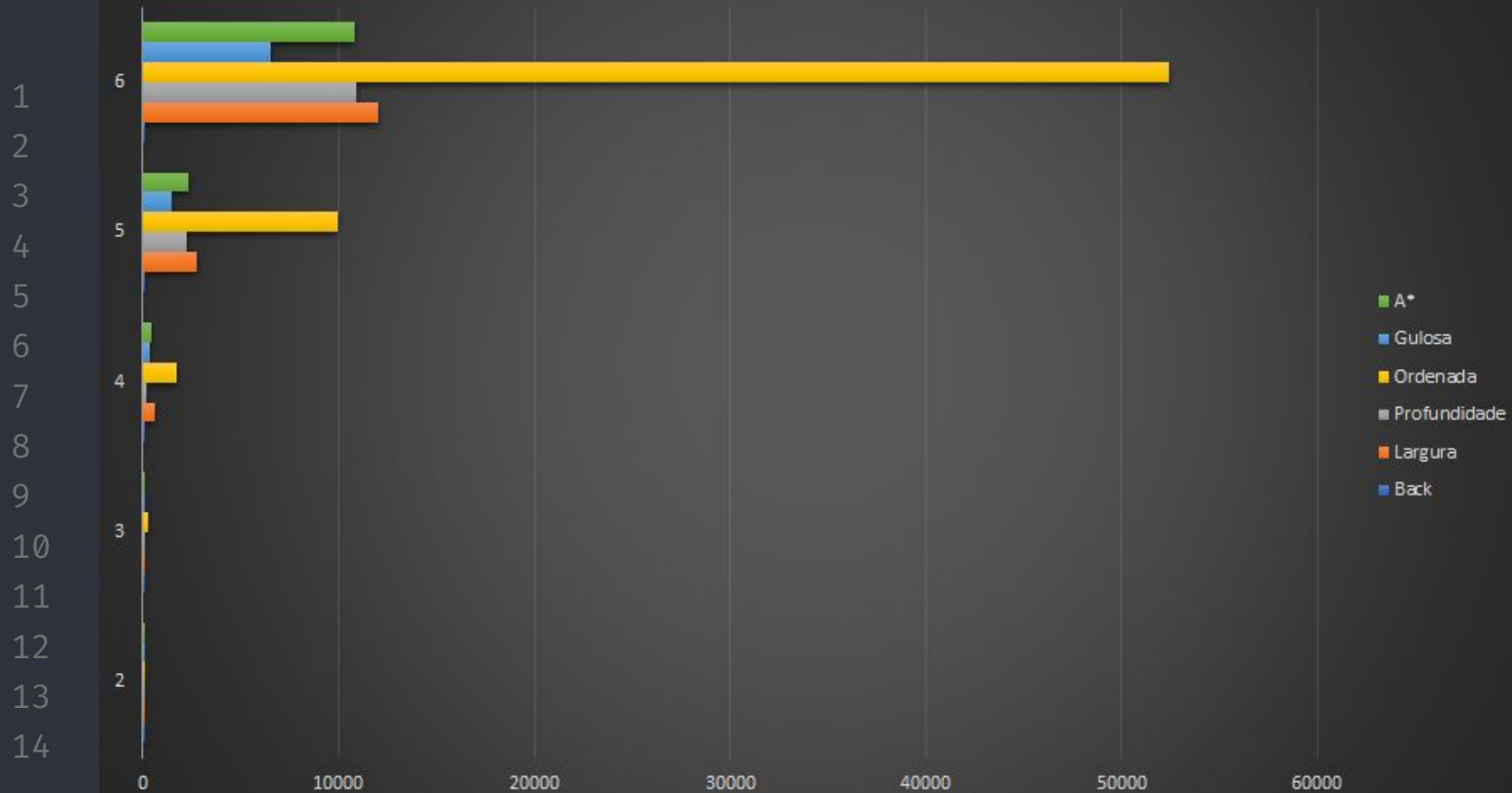
0,0000000000000000 0,2000000000000000 0,4000000000000000 0,6000000000000000 0,8000000000000000 1,0000000000000000 1,2000000000000000

Inteligência Artificial

Nó Visitados



Nó Expandidos



```
1  Dificuldades {
2
3
4
5
6      Referências e Literatura
7
8      Heurística
9
10
11
12
13
14 }
```

```
1  Obrigado; {
2
3      'Dúvidas?'
4
5
6
7      Bom descanso!
8
9
10     CREDITS: This presentation template was
11     created by Slidesgo, including icons by
12     Flaticon, and infographics & images by Freepik
13
14 }
```