



UNIVERSIDADE FEDERAL DE MATO GROSSO.
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO.
DISCIPLINA: INTELIGÊNCIA ARTIFICIAL

RELATÓRIO DE IMPLEMENTAÇÃO
PUZZLE

Discentes:

Guilherme Santana Coelho

Vitor Rezende Campos

Guilherme Henrique Mendes

Docente:

Robson Silva Lopes

Outubro de 2014, Barra do Garças- MT

1.Descrição sucinta das implementações

O projeto foi desenvolvido utilizando a linguagem de programação Python.

Cada jogada é representada por uma classe chamada **noPuzzle** contendo os seguintes atributos:

- Jogada (Estado do tabuleiro)
- Pai (Indica a jogada anterior)
- Filhos (Vetor com as jogadas geradas a partir desta)
- Nivel (Nível da árvore que se encontra o nó)
- Qtdfilhos (Quantidade de Filhos)
- Movimento (Direção do movimento)

Em todos os métodos de busca, serão utilizados como entradas, os casos de teste da tabela a seguir:

Casos de teste:

Meta:	Fácil:	Médio:	Difícil:	Extremamente Difícil:
1 2 3	1 3 4	2 8 1	2 8 1	5 6 7
8 X 4	8 6 2	X 4 3	4 6 3	4 X 8
7 6 5	7 X 5	7 6 5	X 7 5	3 2 1

1.1 Implementação do algoritmo de busca em largura

Na busca em largura a partir do momento que o nó raiz(jogada inicial) é visitado, seus filhos (possibilidades de jogadas posteriores) são gerados e adicionados no **final** de uma lista que é utilizada para que a busca possa ser implementada, se o nó for o objetivo que pretendemos alcançar esse estado é retornado. Caso não seja, a busca prossegue com o primeiro nó dessa lista, verificando-o e gerando seus filhos (próximas jogadas). A lista que utilizamos trabalha como uma **fila** para as jogadas que devem ser verificadas, dando assim a característica da busca em largura.

Para a solução Fácil a saída é:

Solução:	Nós:	Níveis:
1 3 4 8 6 2 7 9 5	0	0
1 3 4 8 9 2 7 6 5	3	1
1 3 4 8 2 9 7 6 5	8	2
1 3 9 8 2 4 7 6 5	18	3
1 9 3 8 2 4 7 6 5	32	4
1 2 3 8 9 4 7 6 5	41	5

O número de nós visitados foram 41. E o nível da solução é o nível 5.

Para a solução Média a saída é:

Solução:	Nós:	Níveis:
2 8 1 9 4 3 7 6 5	0	0
9 8 1 2 4 3 7 6 5	3	1
8 9 1 2 4 3 7 6 5	8	2
8 1 9 2 4 3 7 6 5	18	3
8 1 3 2 4 8 7 6 5	32	4
8 1 3 2 9 4 7 6 5	60	5
8 1 3 9 2 4 7 6 5	102	6
9 1 3 8 2 4 7 6 5	182	7
1 9 3 8 2 4 7 6 5	290	8
1 2 3 8 9 4 7 6 5	294	9

O número de nós visitados foi 294 e o nível em que se encontrou a solução foi no nível 9.

Para a solução Difícil a saída é:

Solução:	Nós:	Níveis:
2 8 1 4 6 3 0 7 5	0	0
2 8 1 4 6 3 7 0 5	2	1
2 8 1 4 0 3 7 6 5	6	2
2 8 1 0 4 3 7 6 5	14	3
0 8 1 2 4 3 7 6 5	30	4
8 0 1 2 4 3 7 6 5	50	5
8 1 0 2 4 3 7 6 5	89	6
8 1 3 2 4 0 7 6 5	151	7
8 1 3 2 0 4 7 6 5	267	8
8 1 3 0 2 4 7 6 5	419	9
0 1 3 8 2 4 7 6 5	705	10
1 0 3 8 2 4 7 6 5	1101	11
1 2 3 8 0 4 7 6 5	1579	12

O número de nós visitados foi 1579 e o nível em que se encontrou a solução foi no nível 12.

Para a solução Extremamente Difícil a saída é:

Para a entrada Extremamente Difícil torna-se inviável encontrar o objetivo. O algoritmo passa de 20 minutos para alcançar pouco mais de 20 níveis.

1.1.1 – Fator de Ramificação

Seja **d** a altura da árvore e **N** o número de nós, então: $N+1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d$, sendo **b*** o fator de ramificação.

Nível do tabuleiro	Fator de Ramificação
Fácil	1,79705
Médio	1,67572
Difícil	1,71335

1.2 Implementação do algoritmo de busca em Profundidade

A única diferença na implementação da busca em profundidade para a busca em largura é que a lista utilizada para guardar os nós que devem ser explorados trabalha como uma **pilha**, inserindo os filhos (novos estados) sempre no **início** da dessa lista e continuando fazendo o teste para o primeiro elemento dela estaremos garantindo que a busca irá sempre descer ao filho de maior profundidade antes de testar os demais.

Para a solução Fácil a saída é:

A solução encontra-se no nível 57. Como a busca em profundidade tende a descer até que um nó gere todos os seus filhos com jogadas repetidas, o número de nós explorados são iguais à 57.

Para a solução Média a saída é:

Solução:	Níveis:
2 8 1 9 4 3 7 6 5	0
9 8 1 2 4 3 7 6 5	1
8 9 1 2 4 3 7 6 5	2
8 1 9 2 4 3 7 6 5	3
8 1 3 2 4 9 7 6 5	4
8 1 3 2 9 4 7 6 5	5
8 9 3 2 1 4 7 6 5	6
9 8 3 2 1 4 7 6 5	7
2 8 3 9 1 4 7 6 5	8
2 8 3 1 9 4 7 6 5	9
2 9 3 1 8 4 7 6 5	10
9 2 3 1 8 4 7 6 5	11
1 2 3 9 8 4 7 6 5	12
1 2 3 8 9 4 7 6 5	13

A solução encontra-se no nível 13. Como a busca em profundidade tende a descer até que um nó gere todos os seus filhos com jogadas repetidas, o número de nós explorados são iguais à 13.

Para a solução Difícil a saída é:

A busca desce mais de 20 mil níveis sem solução encontrada. O mesmo acontece para o caso **Extremamente Difícil**.

1.2.1 – Fator de Ramificação

Seja d a altura da árvore e N o número de nós, então: $N+1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d$, sendo b^* o fator de ramificação.

Nível do tabuleiro	Fator de Ramificação
Fácil	-
Médio	1,76363
Difícil	-

1.3 Implementação do algoritmo de busca em aprofundamento Iterativo

A busca com aprofundamento iterativo é uma combinação da busca em profundidade com a busca em largura, basicamente estabelecemos um limite e a busca vai tentando iterativamente achar a solução com uma busca em profundidade limitada do nível 1 até o nível limite estabelecido. Devido o bloqueio que a linguagem Python faz à chamadas recursivas infinitas, implementamos essa busca com um contador manual de iteração.

Para a solução Fácil a saída é:

Solução:	Nível
1 3 4 8 6 2 7 9 5	0
1 3 4 8 9 2 7 6 5	1
1 3 4 8 2 9 7 6 5	2
1 3 9 8 2 4 7 6 5	3
1 9 3 8 2 4 7 6 5	4
1 2 3 8 9 4 7 6 5	5

O número de nós visitados foram 23. E o nível da solução é o nível 5.

Para a solução Média a saída é:

Solução:	Nível
2 8 1 9 4 3 7 6 5	0
9 8 1 2 4 3 7 6 5	1
8 9 1 2 4 3 7 6 5	2
8 1 9 2 4 3 7 6 5	3
8 1 3 2 4 9 7 6 5	4
2 8 1 9 4 3 7 6 5	5
8 1 3 2 9 4 7 6 5	6
8 1 3 2 9 4 7 6 5	7
1 9 3 8 2 4 7 6 5	8
1 2 3 8 9 4 7 6 5	9

O número de nós visitados foram 18. E o nível da solução é o nível 9.

Para a solução Difícil a saída é:

Solução	Nível
2 8 1 4 6 3 9 7 5	0
2 8 1 9 6 3 4 7 5	1
9 8 1 2 6 3 4 7 5	2
8 9 1 2 6 3 4 7 5	3
8 1 9 2 6 3 4 7 5	4
...	
1 2 3 9 6 4 8 7 5	25
1 2 3 8 6 4 9 7 5	26
1 2 3 8 6 4 7 9 5	27
1 2 3 8 9 4 7 6 5	28

O número de nós visitados foram 2795. E o nível da solução é o nível 28.

1.3.2– Fator de Ramificação

Seja d a altura da árvore e N o número de nós, então: $N+1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d$, sendo b^* o fator de ramificação.

Nível do tabuleiro	Fator de Ramificação
Fácil	1.54277
Médio	1,13638
Difícil	1,2532

2 - Conclusão

Os testes dos algoritmos foram realizados no sistema Linux Mint 17.

O algoritmo nos surpreendeu pois foi capaz de nos trazer soluções para a maioria dos casos e dificuldades, embora tenha algumas limitações, podemos compreender com os testes realizados que elas são trazidas devido ao grande número de possibilidades que as buscas cegas tendem à explorar, acarretando assim no uso de uma grande quantidade de memória e processamento, por isso algumas soluções podem demandar muito tempo para serem obtidas nessas condições.