Aplicando Buscas no jogo Sokoban

Felipe Lana Machado[1]

[1]Engenharia de Software – Universidade do Estado de Santa Catarina (UDESC)
R. Dr. Getúlio Vargas, 2822 - Bela Vista, Ibirama - SC, 89140-000

Abstract. *This work was developed to apply several types of blind and intelligent searches to solve a set of phases in the game sokoban, which aims to put boxes in their proper places. This made a comparison between the searches, highlighting the amount of game states visited and the time spent to find the solution. Still, highlighting if the result found was the best solution or not.*

Resumo. *Este trabalho foi desenvolvido para aplicar diversos tipos de buscas cegas e inteligentes para resolver um conjunto de fases no jogo sokoban, o qual tem como objetivo colocar caixas nos seus devidos lugares. Com isso foi realizada uma comparação entre as buscas, destacando a quantidade de estados do jogo visitados e o tempo gasto para encontrar a solução. Ainda, destacando se o resultado encontrado foi ótimo ou não.*

1. Introdução

Sokoban é um jogo clássico invetado no Japão por volta de 1980, sendo que o jogo original foi escrito por Hiroyuki Imabayashi [SokobanWIKI 2008]. Uma fase do jogo é representada por um galpão onde existem caixas espalhadas. O objetivo é ajudar o organizador (jogador) a passar pelo labirinto que existe no galpão e colocar as caixas nos locais indicados. As caixas devem sempre ser empurradas, não puxadas e somente uma caixa pode ser empurrada por vez.

O jogo *sokoban* é considerado um problema np-difícil [Dor and Zwick 1999], sendo que alguns trabalhos o consideram ainda mais difícil que os problemas np [Culberson 1997]. Resolver o jogo pode ajudar na automação de robôs que movimentam caixas reais em galpões, sendo um estudo recorrente de pesquisadores de inteligência computacional.

Para resolver o jogo podem ser utilizadas buscas que utilizam um único agente de busca e algoritmos como a busca A* e outras buscas cegas, como a busca em largura ou a busca em profundidade. Porém, estas buscas somente não são capazes de resolver problemas mais complexos, sendo que neste trabalho foram utilizadas instâncias mais fáceis de serem resolvidas.

Este trabalho está organizado de forma que a segunda sessão apresenta como foi modelado a representação do estado, seguido da explicação e detalhamento da heurística em uma das buscas utilizadas para resolver o problema proposto. Na terceira sessão, mostra-se as configurações de hardware do computador utilizado para realizar os experimentos. Em sequência, são apresentados os resultados obtidos realizando o experimento com as quatro buscas nos dez mapas propostos. Por fim, o último capítulo apresenta as conclusões deste trabalho e sugestões de trabalhos futuros.

2. Representação de Estado

Para representar o jogo foram utilizadas listas que contém as posições cartesianas (a, b) de cada elemento do tabuleiro, isto é, as paredes, as caixas, os objetivos e o jogador. Cada um destes elementos são também representados como um carácter como demonstrado na tabela 1. O estado inicial é fornecido em um arquivo de texto neste formato para a leitura e o estado final ou meta é aquele que todas as caixas ocupam as posições objetivo.

Tabela 1. Tabela representando o estado do jogo visualmente.

	Element	**Character**	**ASCII Code**
0	Wall	#	0x23
1	Player	@	0x40
2	Player on goal square	+	0x2b
3	Box	$	0x24
4	Box on goal square	*	0x2a
5	Goal square	.	0x2e
6	Floor	(space)	0x20

[SokobanWIKI 2008].

As vizinhanças dos estados são todas as possibilidades de movimentação do jogador, ou seja, o jogador pode se mover em um espaço desocupado ou para uma posição de uma caixa, empurrando ela para a posição seguinte na direção que o jogador moveu. Porém, se a posição que a caixa irá ocupar após ser movida for outra caixa ou uma parede, o jogador não poderá realizar este movimento.

A representação da solução do problema é um vetor contendo as direções das movimentações do jogador da posição inicial até a solução do problema, sendo que ”l”representa *left* ou esquerda, ”r”representa *right* ou direita, ”u”representa *up* ou para cima e ”d”representa *down* ou para baixo.

3. Heurística

A heurística utilizada no trabalho foi a *manhattan distance* que é a diferença absoluta de um par de coordenadas. Assim, é verificada a distância entre o jogador e todas as caixas e entre as caixas e os objetivos.

$$d(a, b) = \sum |b_i - a_i| \tag{1}$$

Para uma heurística ser admissível ela deve ser uma heurística otimista, ou seja, ela deve sempre subestimar o custo de ir do estado atual para o estado objetivo [Norvig and Russell 2004]. Assim, quando calculada a distância não é levado em conta os obstáculos e além disso, várias caixas podem ser armazenadas em um mesmo objetivo, portanto a heurística escolhida é admissível.

Porém, ao se deparar com situações em que a caixa deve ser empurrada para um estado mais distante afim de se encontrar a melhor solução, essa heurística falha. Uma

vez que o algoritmo tende a escolher a menor distancia para o objetivo, quando é necessário ir para o lado contrário ao objetivo faz com que o algoritmo não contemple esta possibilidade, gerando por vezes um resultado sub-ótimo.

4. Ambiente de Simulação

O projeto foi implementado na linguagem *Python* utilizando a versão 3.7, foi utilizada a IDE PyCharm para implementação dos códigos. Os códigos foram executados em um computador com processador Intel(R) Core(TM) i7-7700HQ CPU 2.80GHz, com 32GB de memória RAM, utilizando um sistema operacional Windows 10 PRO.

Para geração das tabelas e gráficos, foi utilizado o *Jupyter notebook*, ferramenta que vem junto da distribuição Anaconda3 e facilita o trabalho com os dados e visualizações. Pois, utiliza ferramentas como *pandas* e *matplotlib* que são ferramentas poderosas na linguagem *python* para geração de visualizações e manipulação de *datasets*.

5. Resultados

Os resultados obtidos foram separados em grupos do mesmo mapa, apresentando o resultado de cada busca para aquele mapa. O resultado consistem em cinco métricas principais e a solução:

- *Node* - são os nós ou estados visitados durante a execução do algoritmo.
- *Repeat* - são os nós visitados que são repetidos e portanto foram retirados da execução.
- *Fringe* - representa os nós mais abaixo na árvore, chamados de folhas ou franjas.
- *Deadlock* - são aqueles estados em que uma caixa está em alguma quina de paredes do mapa.
- *Time* - representa o tempo de execução do algoritmo em segundos. O tempo máximo de execução era de três horas e no caso da busca em profundidade o limite de tempo foi a primeira finalização da busca em profundidade limitada, após o limite de tempo.
- *Answer* - a solução no formato *u (up), d (down), l (left), r (right)*

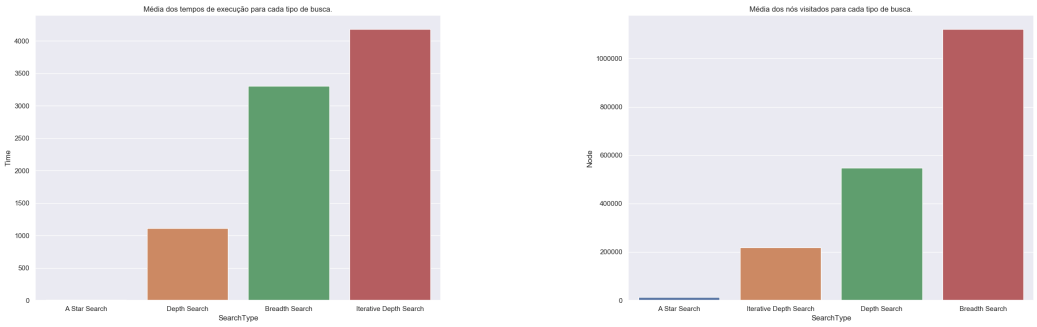
Durante o experimento foi decidido que o tempo máximo de execução de alguma das busca seria de três horas. Assim, é possível verificar que em alguns casos a solução não foi encontrada durante este tempo limite, como pode ser visto na tabela 3, na qual podemos verificar que todos os casos que o tempo de execução foi maior do que 10800 segundos não obtiveram a solução do problema. Estes resultados também podem ser verificados nas tabelas em apêndice.

Tabela 2. Tabela com a média dos resultados obtidos no experimento.

	SearchType	Node	Repeat	Fringe	Deadlock	Time
0	A Star Search	12482.2	5772.0	1965.9	367.8	8.148682
2	Depth Search	547914.8	354130.6	317.9	17352.9	1108.408130
1	Breadth Search	1121904.7	612056.7	61132.9	22018.1	3302.432080
3	Iterative Depth Search	218009.4	116674.1	4049.7	6371.4	4183.690827

Tabela elaborada em 04/04/2019.

Na tabela 2 acima, tem-se as médias das métricas agrupadas pelo tipo de busca e levando em consideração todos os experimentos, inclusive os que não obtiveram sucesso. Com isso, a busca A* que é a busca que utiliza a heurística apresentou a melhor média em quase todas as métricas do problema, apresentando uma média de pouco mais de 8 segundos para resolver os problemas. A busca A* apenas não obteve a melhor métrica em *Fringe* que são os nós folhas da árvore de busca, nesta métrica o menor resultado foi a busca em profundidade, o que é compreensível uma vez que esta busca procura o resultado em apenas um ramo da árvore de busca por vez.



(a) Média do tempo **(b)** Média dos nós visitados

Figura 1. Média de tempo e nós visitados por cada busca

Nos gráficos acima, pode-se perceber que a busca em profundidade iterativa foi a que mais demorou para resolver os problemas, possuindo média de tempo de mais de 4100 segundos. Esta busca foi desenvolvida para armazenar as folhas da execução anterior e continuar a iteração a partir deste ponto, ao invés de começar novamente em todas as interações, o que fez com que a quantidade de nós visitados fosse a menor após a busca A* que ficou com a menor quantidade de nós visitados em média.

Quanto a busca em largura, os resultados também foram esperados, no gráfico da Figura 1 a direita, pode-se perceber que a busca que mais explorou nós foi a busca em largura, como já era o esperado uma vez que esta busca abre todos os nós do problema até encontrar a solução.

Por fim, pode-se destacar que todas as buscas utilizaram a podagem de nós repetidos e também um deadlock simples que é o caso de haver caixas em quinas no mapa. Estas implementações melhoraram a performance de todas as buscas, porém como não há um controle de qual estado é melhor do que outro e apenas o primeiro visitado é armazenado isso pode gerar soluções não ótimas.

Na tabela 3 abaixo, pode-se verificar o resultado de cada busca individualmente para cada mapa proposto como problema neste trabalho, verificando o resultado individual para cada experimento realizado, no final ao apêndice é possível verificar a solução encontrada ou não, individualmente para cada experimento realizado (Tabelas de 4 a 10).

Tabela 3. Tabela com os resultados obtidos no experimento deste projeto.

	SearchType	Screen	Node	Repeat	Fringe	Deadlock	Time
0	Breadth Search	0	663	310	135	3	0.208443
1	Depth Search	0	1748	1172	15	6	0.626342
2	Iterative Depth Search	0	144	80	19	3	0.176578
3	A Star Search	0	38	7	21	0	0.011969
4	Breadth Search	1	56144	33094	2434	383	22.263965
5	Depth Search	1	10292	6283	309	99	5.731885
6	Iterative Depth Search	1	782	408	2601	13	21.144708
7	A Star Search	1	751	317	152	13	0.292795
8	Breadth Search	2	57899	35019	921	1198	20.514968
9	Depth Search	2	28675	17841	146	723	13.022484
10	Iterative Depth Search	2	4692	2609	1993	165	28.523564
11	A Star Search	2	5928	2663	1007	229	2.161283
12	Breadth Search	3	516	244	33	24	0.153621
13	Depth Search	3	776	412	12	38	0.240329
14	Iterative Depth Search	3	49	22	85	4	0.354059
15	A Star Search	3	55	17	11	6	0.016921
16	Breadth Search	4	44256	25592	68	1690	16.180267
17	Depth Search	4	23924	13928	67	1087	9.479948
18	Iterative Depth Search	4	2862	1450	2386	47	82.299430
19	A Star Search	4	9033	4387	814	515	3.513055
20	Breadth Search	5	1620959	680825	381966	47736	10802.937138
21	Depth Search	5	5046017	3280266	1518	164196	10800.132611
22	Iterative Depth Search	5	1441593	740025	0	43636	17533.332137
23	A Star Search	5	62079	25580	14622	1768	45.907016
24	Breadth Search	6	5109855	2947021	35352	73452	10800.425630
25	Depth Search	6	78544	45815	579	1969	53.223569
26	Iterative Depth Search	6	132533	76010	15172	2330	10463.692476
27	A Star Search	6	23844	12887	1087	673	13.755742
28	Breadth Search	7	812153	471165	6044	10964	559.560769
29	Depth Search	7	79019	46908	334	1513	68.255979
30	Iterative Depth Search	7	14254	7978	18211	349	1936.139604
31	A Star Search	7	17878	9358	1238	381	10.574090
32	Breadth Search	8	460	219	33	30	0.139620
33	Depth Search	8	176	88	9	10	0.051862
34	Iterative Depth Search	8	30	11	30	2	0.127662
35	A Star Search	8	60	16	15	9	0.017952
36	Breadth Search	9	3516142	1927078	184343	84701	10801.936382
37	Depth Search	9	209977	128593	190	3888	133.316292
38	Iterative Depth Search	9	583155	338148	0	17165	11771.118057
39	A Star Search	9	5156	2488	692	84	5.235994

Tabela elaborada em 04/04/2019.

6. Conclusões

Este relatório apresentou como quatro buscas performaram em relação a dez mapas do jogo *sokoban*, sendo as buscas em largura, profundidade e profundidade iterativa como buscas cegas e a busca A* utilizando a heurística de *manhattan* como busca informada. Todas as buscas ignoraram estados repetidos e *deadlocks* de quina na sua execução, alguns algoritmos encontram solução para todos os mapas propostos e outros não, tendo como tempo limite para execução 3 horas.

Com isso, fica evidente que o melhor método é o da busca informada A*, utilizando a heurística de *manhattan*. Sendo que esta busca foi a única a solucionar todos os mapas propostos e ainda gastou menos de 46 segundos para resolver o mapa considerado mais difícil, enquanto buscas cegas como largura não conseguiram encontrar a solução em 3 horas de execução.

Portanto, uma continuação possível deste trabalho seria uma melhora específica no algoritmo A*, pesquisando novas heurísticas possíveis e também implementando todas as formas de *deadlock* possíveis. Assim, seria possível realizar os testes deste algoritmo melhorado, em instancias consideradas mais difíceis.

Referências

Culberson, J. (1997). Sokoban is pspace-complete.

Dor, D. and Zwick, U. (1999). Sokoban and other motion planning problems. *Computational Geometry*, 13(4):215–228.

Norvig, P. and Russell, S. (2004). Inteligência artificial. *Editora Campus*, 20.

SokobanWIKI (2008). http://www.sokobano.de/wiki/.

Tabela 4. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
0	Breadth Search	0	['u', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'r']
1	Depth Search	0	['l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'r', 'r']
2	Iterative Depth Search	0	['l', 'l', 'u', 'l', 'l', 'd', 'r', 'r', 'r']
3	A Star Search	0	['l', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'r']
4	Breadth Search	1	['l', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'd', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'r']
5	Depth Search	1	['l', 'l', 'd', 'l', 'u', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'l', 'd', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'r', 'r', 'r', 'r', 'r', 'u', 'l', 'u', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'r', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'd', 'r', 'd', 'd', 'l', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'd', 'r', 'u', 'r', 'r', 'd', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'u', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'r', 'r', 'l', 'l', 'u', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'l', 'l', 'd', 'd', 'l', 'u', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'u', 'd', 'l', 'd', 'l', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'u', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'u', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'u', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'u', 'u', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'd', 'r', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'u', 'u', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'd', 'l', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'd', 'l', 'l', 'u', 'u', 'l', 'u', 'r', 'l', 'd', 'l', 'l', 'u', 'r']
6	Iterative Depth Search	1	['l', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'd', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'r']
7	A Star Search	1	['l', 'u', 'u', 'u', 'd', 'd', 'd', 'd', 'l', 'u', 'l', 'u', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'u', 'u', 'l', 'l', 'd', 'd', 'd', 'd', 'r', 'r', 'u', 'l', 'd', 'l', 'u', 'u', 'd', 'r', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'l', 'l', 'u', 'r']

Tabela elaborada em 04/04/2019.

Tabela 5. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
8	Breadth Search	2	['u', 'l', 'l', 'd', 'l', 'd', 'r', 'u', 'u', 'r', 'r', 'd', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'u', 'r', 'u', 'l']
9	Depth Search	2	['d', 'l', 'l', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'd', 'l', 'l', 'd', 'r', 'r', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'd', 'l', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'r', 'd', 'd', 'l', 'u', 'l', 'l', 'r', 'r', 'd', 'r', 'u', 'u', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'd', 'l', 'l', 'd', 'r', 'r', 'l', 'l', 'u', 'r', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'u', 'l', 'l', 'r', 'r', 'd', 'r', 'u', 'u', 'u', 'l', 'l', 'd', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'u', 'd', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'd', 'r', 'u', 'u', 'l', 'l', 'u', 'l', 'l', 'l', 'd', 'r', 'r', 'd', 'r', 'l', 'u', 'l', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'l', 'r', 'r', 'u', 'l', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'l', 'u', 'r', 'u', 'r', 'r', 'd', 'l', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'l', 'l', 'r', 'r', 'd', 'r', 'u', 'u', 'l', 'l', 'd', 'r', 'd', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'r', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'd', 'r', 'r', 'd', 'd', 'l', 'u', 'l', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'r', 'd', 'l']
10	Iterative Depth Search	2	['u', 'l', 'l', 'd', 'l', 'd', 'r', 'u', 'u', 'r', 'r', 'd', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'u', 'r', 'u', 'l']
11	A Star Search	2	['u', 'l', 'l', 'd', 'l', 'd', 'r', 'u', 'u', 'r', 'r', 'd', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'u', 'l', 'u', 'u', 'r', 'r', 'd', 'l']
12	Breadth Search	3	['d', 'l', 'l', 'u', 'd', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'd', 'd', 'l', 'u']
13	Depth Search	3	['d', 'l', 'l', 'l', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'd', 'r', 'd', 'd', 'l', 'u']
14	Iterative Depth Search	3	['d', 'l', 'l', 'u', 'd', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'd', 'd', 'l', 'u']
15	A Star Search	3	['d', 'l', 'l', 'u', 'd', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'd', 'd', 'l', 'u']
16	Breadth Search	4	['r', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'l', 'l', 'd', 'r', 'r', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'u', 'l', 'd']
17	Depth Search	4	['l', 'd', 'r', 'l', 'u', 'r', 'u', 'r', 'r', 'd', 'd', 'l', 'r', 'r', 'r', 'u', 'l', 'd', 'l', 'l', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'r', 'r', 'r', 'u', 'l', 'd', 'r', 'u', 'u', 'l', 'd', 'd', 'l', 'l', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'r', 'u', 'l', 'l', 'u', 'u', 'r', 'd', 'r', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'l', 'd', 'r', 'r', 'l', 'l', 'u', 'r', 'r', 'u', 'u', 'l', 'd', 'r', 'd', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'r', 'r', 'l', 'l', 'l', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'd', 'r', 'u', 'u', 'l', 'l', 'l', 'u', 'l', 'd']
18	Iterative Depth Search	4	['r', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'l', 'l', 'd', 'r', 'r', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'u', 'l', 'd']
19	A Star Search	4	['r', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'l', 'd', 'l', 'd', 'r', 'r', 'r', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'u', 'l', 'd']

Tabela elaborada em 04/04/2019.

Tabela 6. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
20	Breadth Search	5	solution not found
21	Depth Search	5	solution not found
22	Iterative Depth Search	5	solution not found
23	A Star Search	5	['l', 'l', 'l', 'd', 'r', 'l', 'l', 'd', 'd', 'r', 'r', 'l', 'l', 'u', 'u', 'r', 'u', 'u', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'l', 'd', 'r', 'u', 'l', 'u', 'u', 'r', 'r', 'd', 'r', 'r', 'u', 'l', 'r', 'r', 'd', 'l', 'l', 'u', 'u', 'l', 'l', 'd', 'r', 'r', 'r', 'u', 'r', 'd', 'l', 'd', 'l', 'l', 'u', 'l', 'l', 'd', 'd', 'r', 'u', 'l', 'u', 'r', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'd', 'r', 'd', 'l', 'r', 'u', 'r', 'r', 'd', 'l', 'r', 'd', 'd', 'l', 'l', 'r', 'u', 'u', 'r', 'u', 'l', 'u', 'l', 'l', 'd', 'd', 'r', 'l', 'u', 'u', 'r', 'r', 'd', 'l', 'd', 'l', 'l', 'u', 'u', 'r', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'u', 'l', 'u', 'r', 'u', 'r', 'r', 'd', 'r', 'r', 'd', 'l', 'u', 'l', 'u', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'u', 'u', 'r', 'd', 'u', 'l', 'l', 'u', 'u', 'r', 'd', 'u', 'r', 'd', 'd', 'u', 'l', 'l', 'd', 'r']

Tabela elaborada em 04/04/2019.

Tabela 7. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
24	Breadth Search	6	solution not found
25	Depth Search	6	['r', 'u', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'u', 'l', 'd', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'u', 'r', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'r', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'r', 'l', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'l', 'u', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'd', 'r', 'u', 'u', 'l', 'u', 'u', 'r', 'd', 'l', 'd', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'l', 'd', 'l', 'l', 'r', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'r', 'r', 'r', 'd', 'r', 'u', 'u', 'l', 'u', 'u', 'r', 'd', 'd', 'l', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'l', 'd', 'l', 'l', 'r', 'u', 'u', 'r', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'r', 'r', 'u', 'l', 'l', 'd', 'd', 'l', 'l', 'l', 'r', 'd', 'l', 'd', 'd', 'l', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'r', 'r', 'u', 'l', 'l', 'd', 'd', 'l', 'd', 'l', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'r', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'u', 'd', 'l', 'l', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'd', 'd', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'u', 'd', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'r', 'r', 'u', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'u', 'r', 'l', 'd', 'l', 'l', 'l', 'u', 'l', 'u', 'u', 'r', 'd', 'd', 'd', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'r', 'r', 'u', 'l', 'l', 'u', 'l', 'd', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'r', 'r', 'l', 'l', 'u', 'u', 'l', 'd', 'r', 'd', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u']

Tabela elaborada em 04/04/2019.

Tabela 8. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
26	Iterative Depth Search	6	['u', 'r', 'u', 'u', 'r', 'u', 'u', 'l', 'd', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'r', 'r', 'r', 'u', 'r', 'r', 'r', 'd', 'l', 'l', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'u', 'r', 'd', 'd', 'r', 'd', 'l', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'd', 'd', 'd', 'd', 'r', 'r', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'u', 'l', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u']
27	A Star Search	6	['r', 'u', 'u', 'u', 'r', 'u', 'u', 'r', 'r', 'd', 'l', 'u', 'l', 'l', 'd', 'l', 'l', 'r', 'r', 'u', 'r', 'r', 'd', 'l', 'l', 'l', 'r', 'u', 'u', 'u', 'r', 'd', 'l', 'd', 'd', 'r', 'r', 'u', 'l', 'd', 'l', 'u', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'd', 'l', 'l', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'd', 'd', 'r', 'd', 'l', 'l', 'r', 'd', 'd', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'u', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'd', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'u', 'd', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'u', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'd', 'r', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'd', 'd', 'r', 'd', 'l', 'l', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'u', 'u', 'l', 'l', 'r', 'r', 'd', 'd', 'l', 'l', 'l', 'u', 'u', 'u']
28	Breadth Search	7	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'd', 'u', 'u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'r', 'd', 'd', 'd', 'l', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l']

Tabela elaborada em 04/04/2019.

Tabela 9. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
29	Depth Search	7	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'l', 'l', 'u', 'r', 'u', 'r', 'd', 'l', 'u', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'l', 'u', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'd', 'l', 'l', 'u', 'u', 'r', 'd', 'l', 'u', 'r', 'u', 'r', 'd', 'l', 'u', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'l', 'u', 'r', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'd', 'l', 'l', 'u', 'u', 'r', 'd', 'l', 'u', 'r', 'u', 'r', 'd', 'l', 'u', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'd', 'l', 'l', 'u', 'u', 'r', 'd', 'r', 'u', 'l', 'l', 'd', 'd', 'd', 'r', 'u', 'l', 'u', 'r', 'l', 'd', 'd', 'r', 'u', 'r', 'u', 'l', 'l', 'u', 'r', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'l', 'l', 'u', 'r', 'u', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'd', 'l', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'u', 'r', 'u', 'r', 'u', 'u', 'l', 'u', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'u', 'l', 'u', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l']
30	Iterative Depth Search	7	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'd', 'u', 'u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'l', 'd', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'r', 'u', 'u', 'l', 'u', 'u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'r', 'r', 'u', 'l', 'd', 'l', 'l', 'l', 'l', 'd', 'r', 'd', 'd', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'u', 'l', 'u', 'r', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'l', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l']

Tabela elaborada em 04/04/2019.

Tabela 10. Tabela com algumas soluções obtidas no experimento deste projeto.

	SearchType	Screen	Answer
31	A Star Search	7	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'd', 'l', 'd', 'd', 'r', 'u', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'r', 'r', 'u', 'r', 'r', 'd', 'l', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'u', 'd', 'd', 'd', 'd', 'd', 'r', 'u', 'l', 'u', 'u', 'r', 'r', 'u', 'r', 'u', 'u', 'u', 'l', 'l', 'd', 'r', 'u', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'r', 'u', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'd', 'l', 'r', 'u', 'r', 'r', 'd', 'l', 'd', 'd', 'l', 'd', 'l', 'd', 'l', 'u', 'u', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'u', 'l', 'l', 'd', 'r', 'u', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'r', 'l', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'l', 'd', 'r', 'u', 'u', 'r', 'u', 'r', 'r', 'u', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'd', 'd', 'd', 'r', 'd', 'd', 'l', 'u', 'u', 'u', 'u', 'l', 'u', 'r', 'l', 'u', 'u', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'r', 'u', 'u', 'l', 'l', 'l', 'd', 'l', 'd', 'd', 'r', 'd', 'd', 'd', 'r', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'u', 'u', 'r']
32	Breadth Search	8	['d', 'l', 'd', 'd', 'r', 'u', 'r', 'r', 'd', 'l', 'd', 'l', 'u']
33	Depth Search	8	['l', 'd', 'd', 'd', 'r', 'd', 'r', 'u', 'l', 'u', 'l', 'd', 'r', 'd', 'r', 'u', 'r', 'u', 'l']
34	Iterative Depth Search	8	['d', 'l', 'd', 'd', 'r', 'u', 'd', 'd', 'r', 'u', 'r', 'u', 'l']
35	A Star Search	8	['d', 'l', 'd', 'd', 'r', 'u', 'r', 'r', 'd', 'l', 'd', 'l', 'u']
36	Breadth Search	9	solution not found
37	Depth Search	9	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'l', 'l', 'r', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'l', 'r', 'u', 'u', 'l', 'u', 'l', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'r', 'r', 'd', 'r', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'l', 'l', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'd', 'r', 'r', 'r', 'd', 'l', 'd', 'd', 'l', 'l', 'u', 'r', 'u', 'r', 'u', 'u', 'r', 'u', 'l', 'l', 'd', 'r', 'd', 'd', 'l', 'd', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'r', 'd', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'd', 'r', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'd', 'l', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'u', 'l', 'u', 'u', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'l', 'r', 'd', 'l', 'u', 'r', 'u', 'l', 'l', 'l', 'd', 'r', 'l', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'd', 'r', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'u', 'l', 'd', 'l', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'u', 'd', 'd', 'l', 'd', 'l', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'u', 'u', 'u', 'u']
38	Iterative Depth Search	9	solution not found
39	A Star Search	9	['l', 'l', 'l', 'l', 'd', 'd', 'd', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'l', 'u', 'l', 'u', 'l', 'l', 'd', 'r', 'd', 'r', 'l', 'u', 'u', 'r', 'r', 'd', 'r', 'd', 'l', 'l', 'r', 'u', 'u', 'l', 'l', 'd', 'r', 'd', 'r', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'l', 'u', 'r', 'd', 'd', 'r', 'r', 'u', 'u', 'u', 'u', 'r', 'u', 'l', 'd', 'l', 'u', 'd', 'r', 'd', 'd', 'd', 'd', 'l', 'l', 'u', 'r', 'u', 'l', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'u', 'u', 'u', 'r', 'u', 'r', 'd', 'r', 'd', 'd', 'l', 'd', 'l', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'u', 'r', 'u', 'l', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'r', 'u', 'l', 'u', 'l', 'd', 'l', 'l', 'u', 'r', 'l', 'd', 'r', 'r', 'r', 'd', 'd', 'd', 'l', 'd', 'l', 'l', 'u', 'r', 'r', 'l', 'l', 'd', 'r', 'd', 'r', 'r', 'u', 'u', 'u', 'u', 'r']

Tabela elaborada em 04/04/2019.