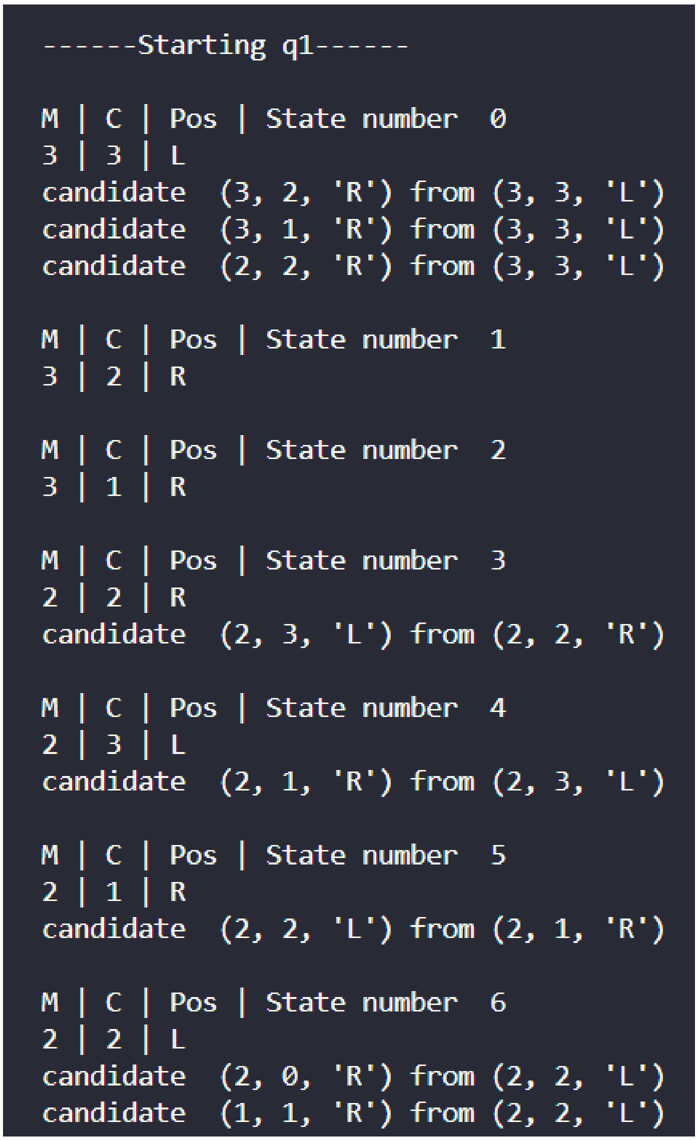
Trabalho 1 de Inteligência Artificial

João Carlos Zucchi

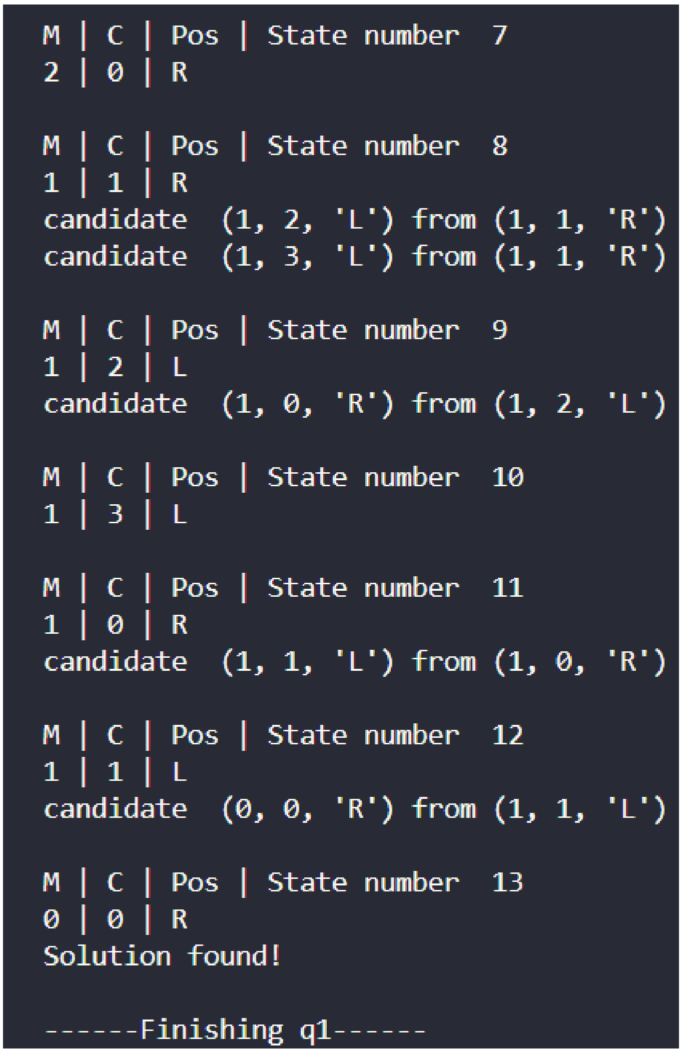
Prof Luis Alvaro Silva

Abril de 2024

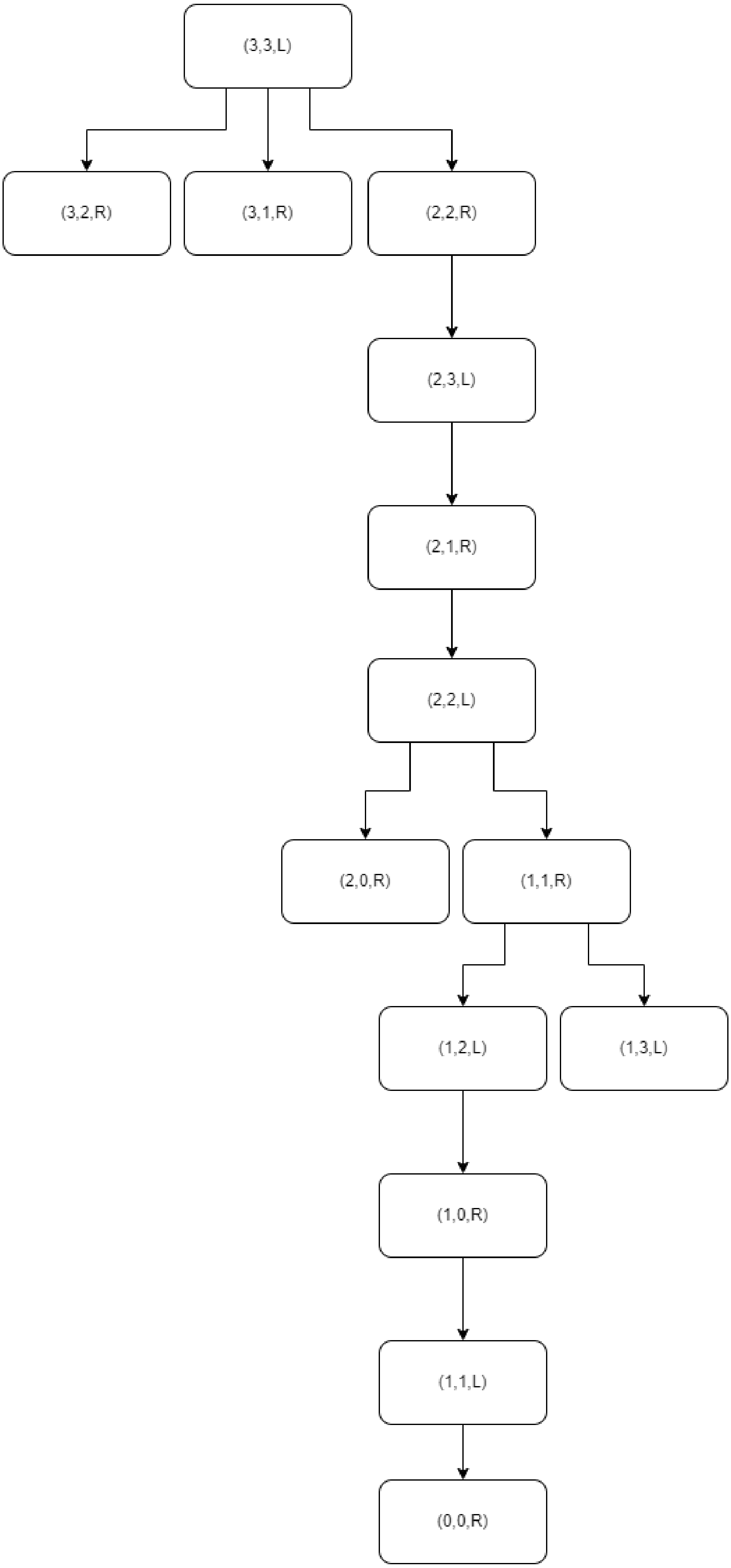
Questão 1:



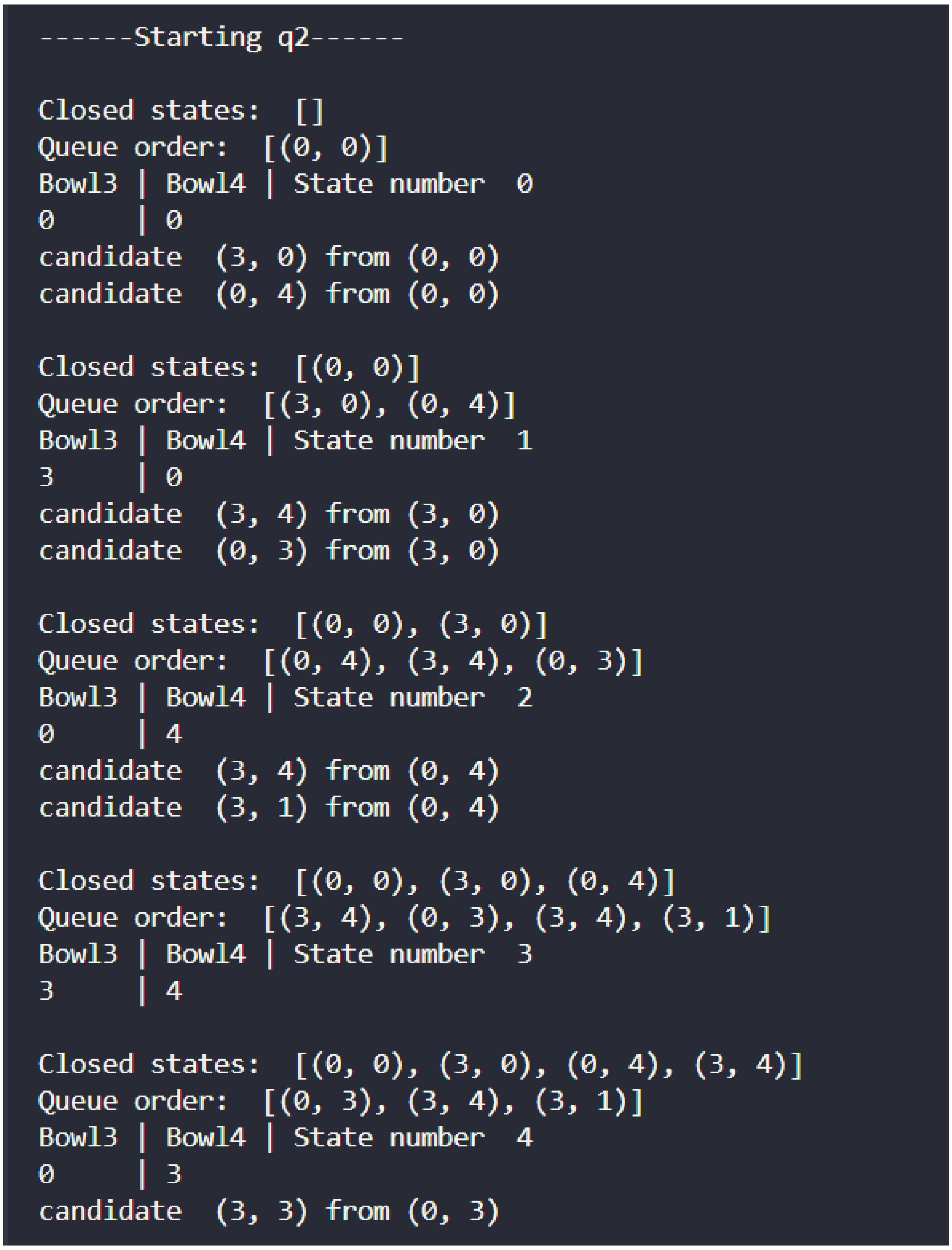
```
------Starting q1------

M | C | Pos | State number  0
3 | 3 | L
candidate  (3, 2, 'R') from (3, 3, 'L')
candidate  (3, 1, 'R') from (3, 3, 'L')
candidate  (2, 2, 'R') from (3, 3, 'L')

M | C | Pos | State number  1
3 | 2 | R

M | C | Pos | State number  2
3 | 1 | R

M | C | Pos | State number  3
2 | 2 | R
candidate  (2, 3, 'L') from (2, 2, 'R')

M | C | Pos | State number  4
2 | 3 | L
candidate  (2, 1, 'R') from (2, 3, 'L')

M | C | Pos | State number  5
2 | 1 | R
candidate  (2, 2, 'L') from (2, 1, 'R')

M | C | Pos | State number  6
2 | 2 | L
candidate  (2, 0, 'R') from (2, 2, 'L')
candidate  (1, 1, 'R') from (2, 2, 'L')
```



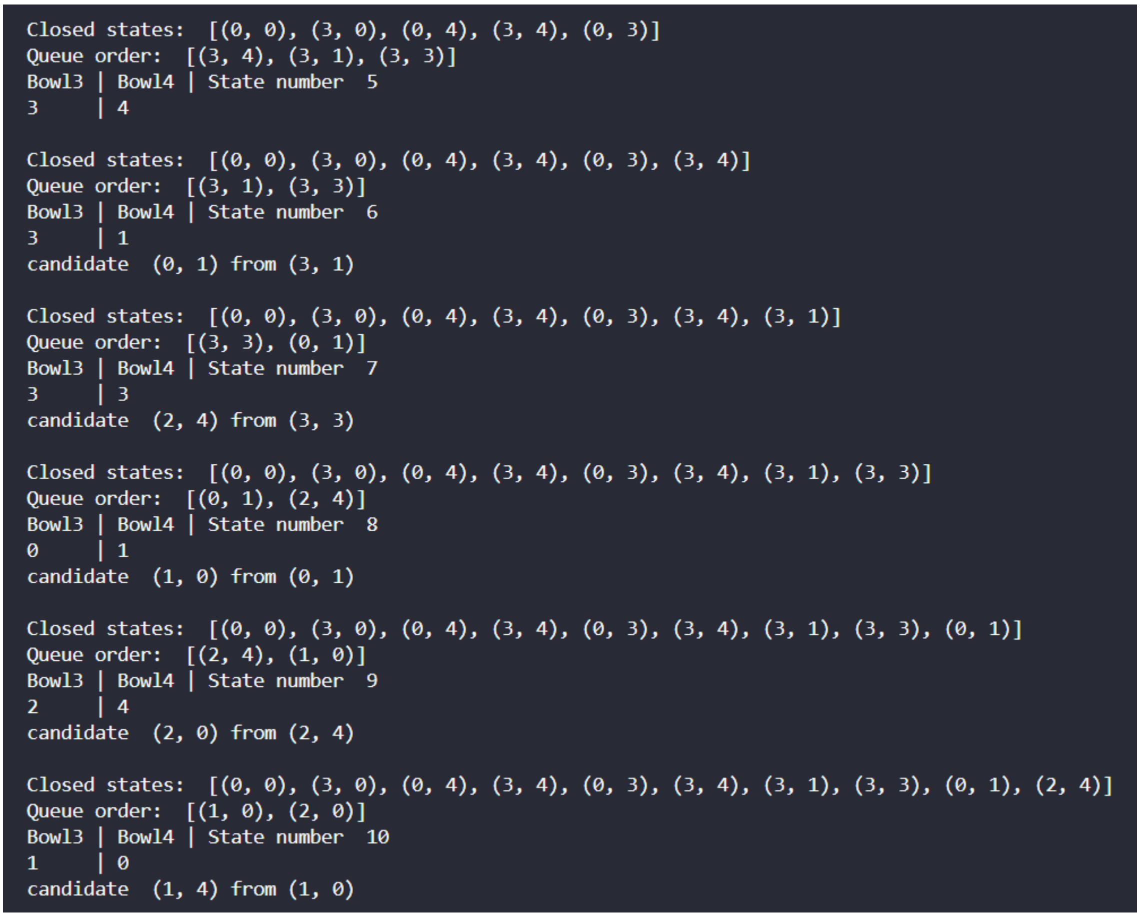
```
M | C | Pos | State number  7
2 | 0 | R

M | C | Pos | State number  8
1 | 1 | R
candidate  (1, 2, 'L') from (1, 1, 'R')
candidate  (1, 3, 'L') from (1, 1, 'R')

M | C | Pos | State number  9
1 | 2 | L
candidate  (1, 0, 'R') from (1, 2, 'L')

M | C | Pos | State number  10
1 | 3 | L

M | C | Pos | State number  11
1 | 0 | R
candidate  (1, 1, 'L') from (1, 0, 'R')

M | C | Pos | State number  12
1 | 1 | L
candidate  (0, 0, 'R') from (1, 1, 'L')

M | C | Pos | State number  13
0 | 0 | R
Solution found!

------Finishing q1------
```


(3,3,L)
(3,2,R)
(3,1,R)
(2,2,R)
(2,3,L)
(2,1,R)
(2,2,L)
(2,0,R)
(1,1,R)
(1,2,L)
(1,3,L)
(1,0,R)
(1,1,L)
(0,0,R)

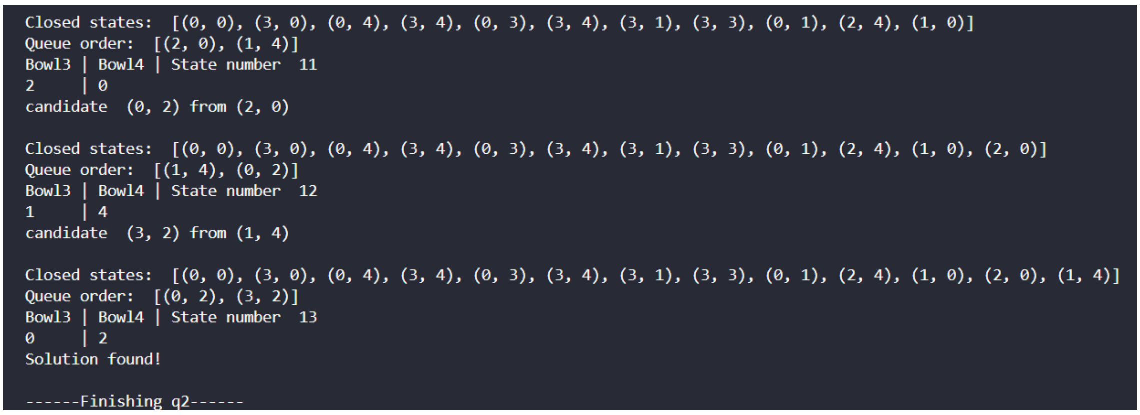
Questão 2:



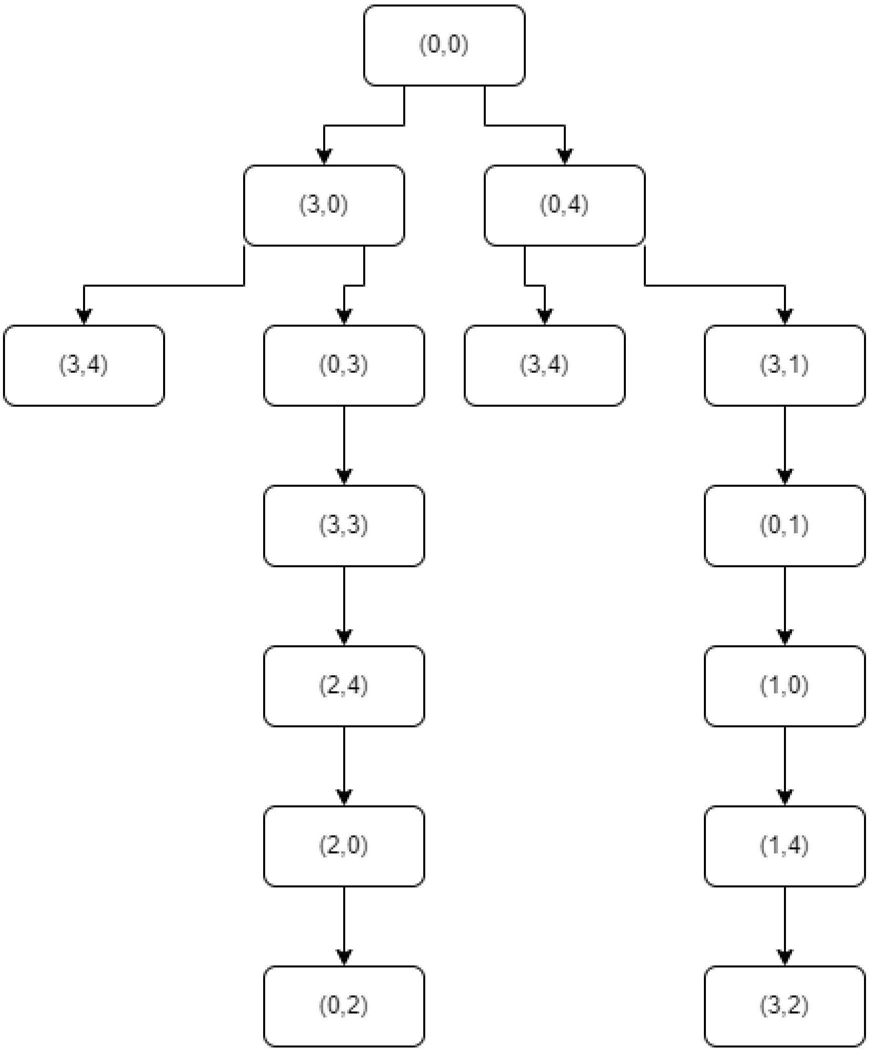
```
------Starting q2------

Closed states:  []
Queue order:  [(0, 0)]
Bowl3 | Bowl4 | State number  0
0     | 0
candidate  (3, 0) from (0, 0)
candidate  (0, 4) from (0, 0)

Closed states:  [(0, 0)]
Queue order:  [(3, 0), (0, 4)]
Bowl3 | Bowl4 | State number  1
3     | 0
candidate  (3, 4) from (3, 0)
candidate  (0, 3) from (3, 0)

Closed states:  [(0, 0), (3, 0)]
Queue order:  [(0, 4), (3, 4), (0, 3)]
Bowl3 | Bowl4 | State number  2
0     | 4
candidate  (3, 4) from (0, 4)
candidate  (3, 1) from (0, 4)

Closed states:  [(0, 0), (3, 0), (0, 4)]
Queue order:  [(3, 4), (0, 3), (3, 4), (3, 1)]
Bowl3 | Bowl4 | State number  3
3     | 4

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4)]
Queue order:  [(0, 3), (3, 4), (3, 1)]
Bowl3 | Bowl4 | State number  4
0     | 3
candidate  (3, 3) from (0, 3)
```



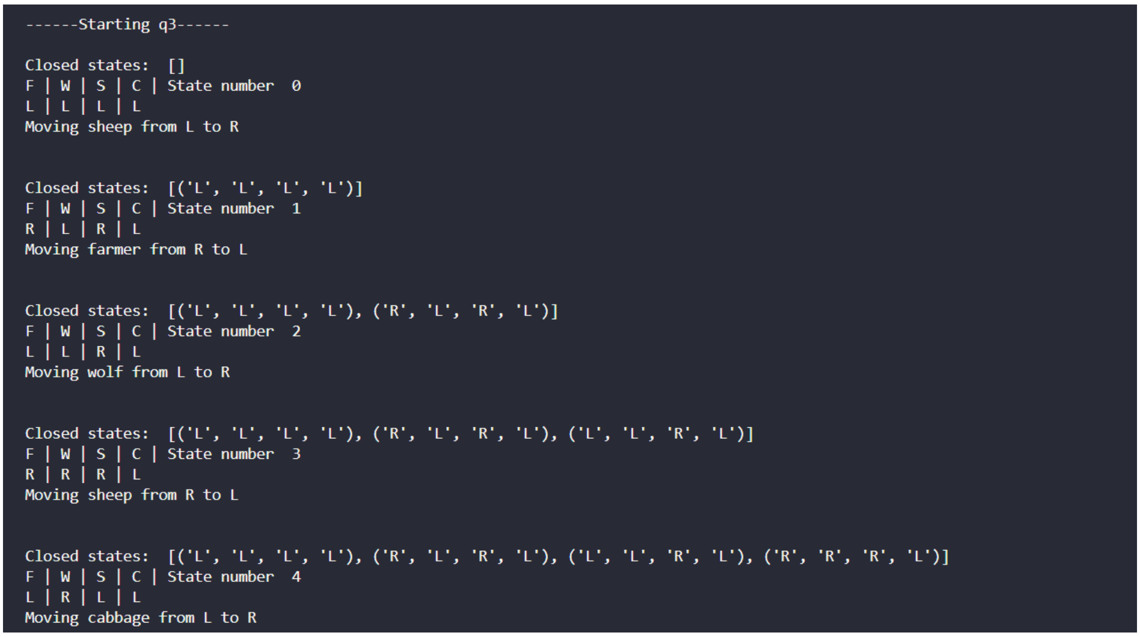
```
Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3)]
Queue order:  [(3, 4), (3, 1), (3, 3)]
Bowl3 | Bowl4 | State number  5
3      | 4

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4)]
Queue order:  [(3, 1), (3, 3)]
Bowl3 | Bowl4 | State number  6
3      | 1
candidate  (0, 1) from (3, 1)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1)]
Queue order:  [(3, 3), (0, 1)]
Bowl3 | Bowl4 | State number  7
3      | 3
candidate  (2, 4) from (3, 3)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3)]
Queue order:  [(0, 1), (2, 4)]
Bowl3 | Bowl4 | State number  8
0      | 1
candidate  (1, 0) from (0, 1)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3), (0, 1)]
Queue order:  [(2, 4), (1, 0)]
Bowl3 | Bowl4 | State number  9
2      | 4
candidate  (2, 0) from (2, 4)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3), (0, 1), (2, 4)]
Queue order:  [(1, 0), (2, 0)]
Bowl3 | Bowl4 | State number  10
1      | 0
candidate  (1, 4) from (1, 0)
```



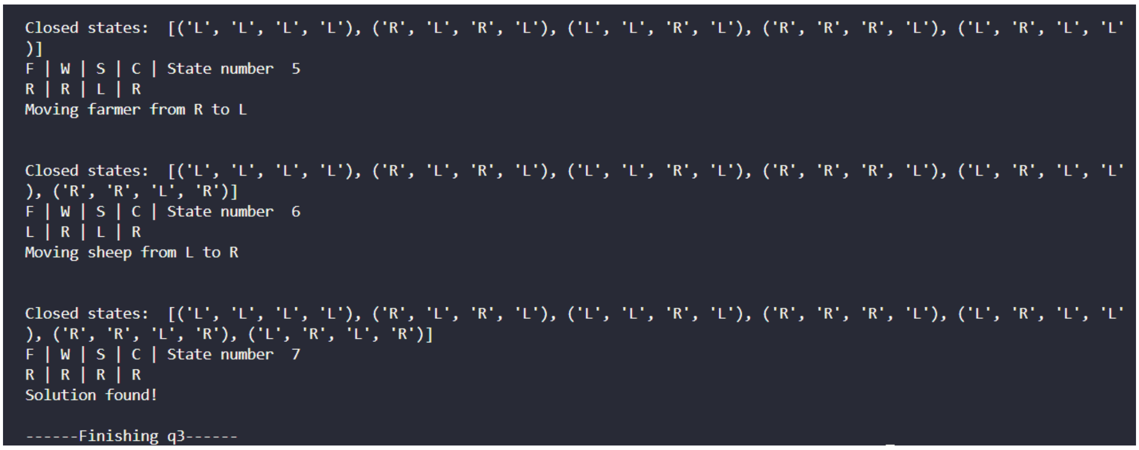
```
Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3), (0, 1), (2, 4), (1, 0)]
Queue order:  [(2, 0), (1, 4)]
Bowl3 | Bowl4 | State number  11
2      | 0
candidate  (0, 2) from (2, 0)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3), (0, 1), (2, 4), (1, 0), (2, 0)]
Queue order:  [(1, 4), (0, 2)]
Bowl3 | Bowl4 | State number  12
1      | 4
candidate  (3, 2) from (1, 4)

Closed states:  [(0, 0), (3, 0), (0, 4), (3, 4), (0, 3), (3, 4), (3, 1), (3, 3), (0, 1), (2, 4), (1, 0), (2, 0), (1, 4)]
Queue order:  [(0, 2), (3, 2)]
Bowl3 | Bowl4 | State number  13
0      | 2
Solution found!

------Finishing q2------
```


(0,0)
(3,0)
(0,4)
(3,4)
(0,3)
(3,4)
(3,1)
(3,3)
(0,1)
(2,4)
(1,0)
(2,0)
(1,4)
(0,2)
(3,2)

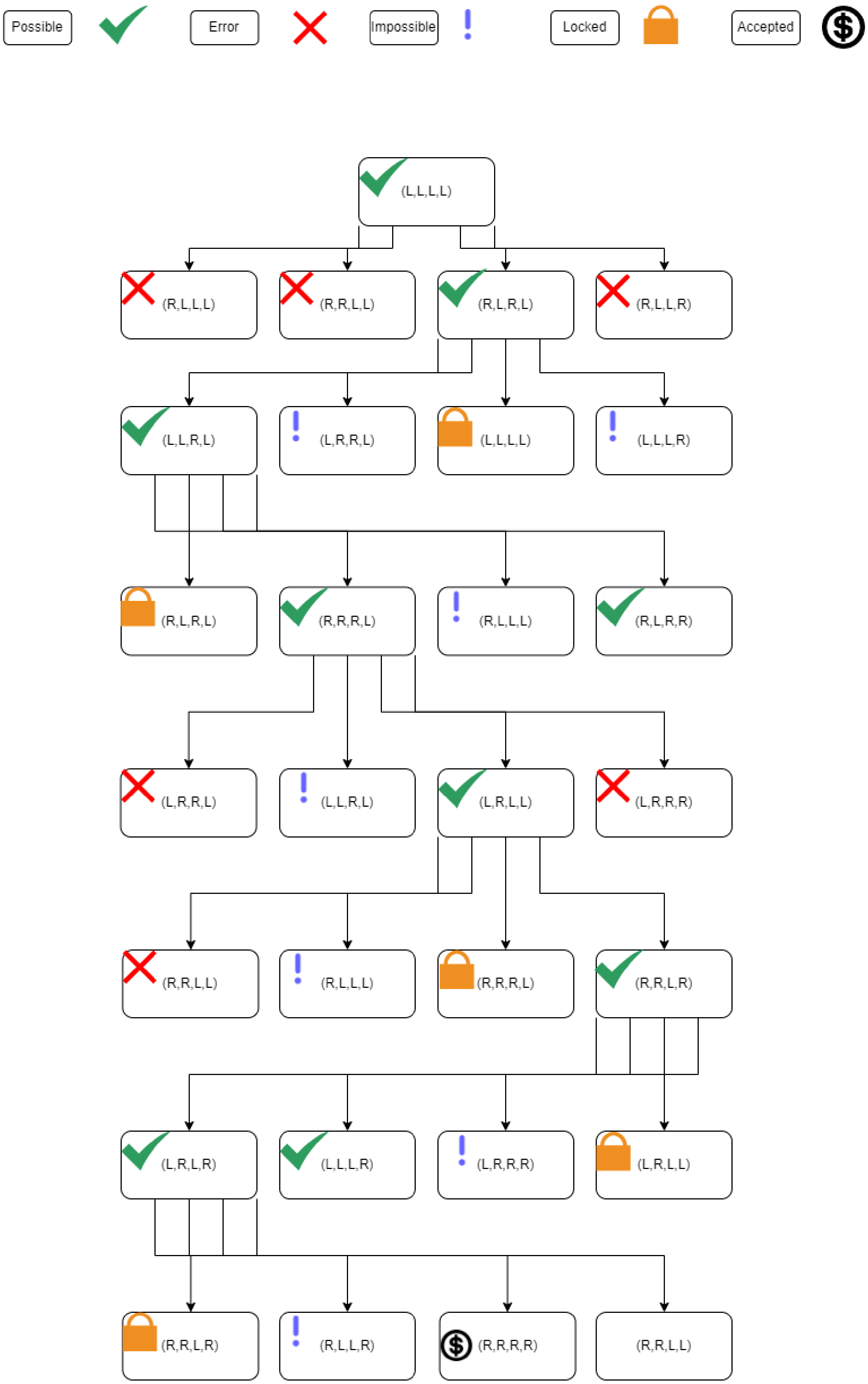
Questão 3:



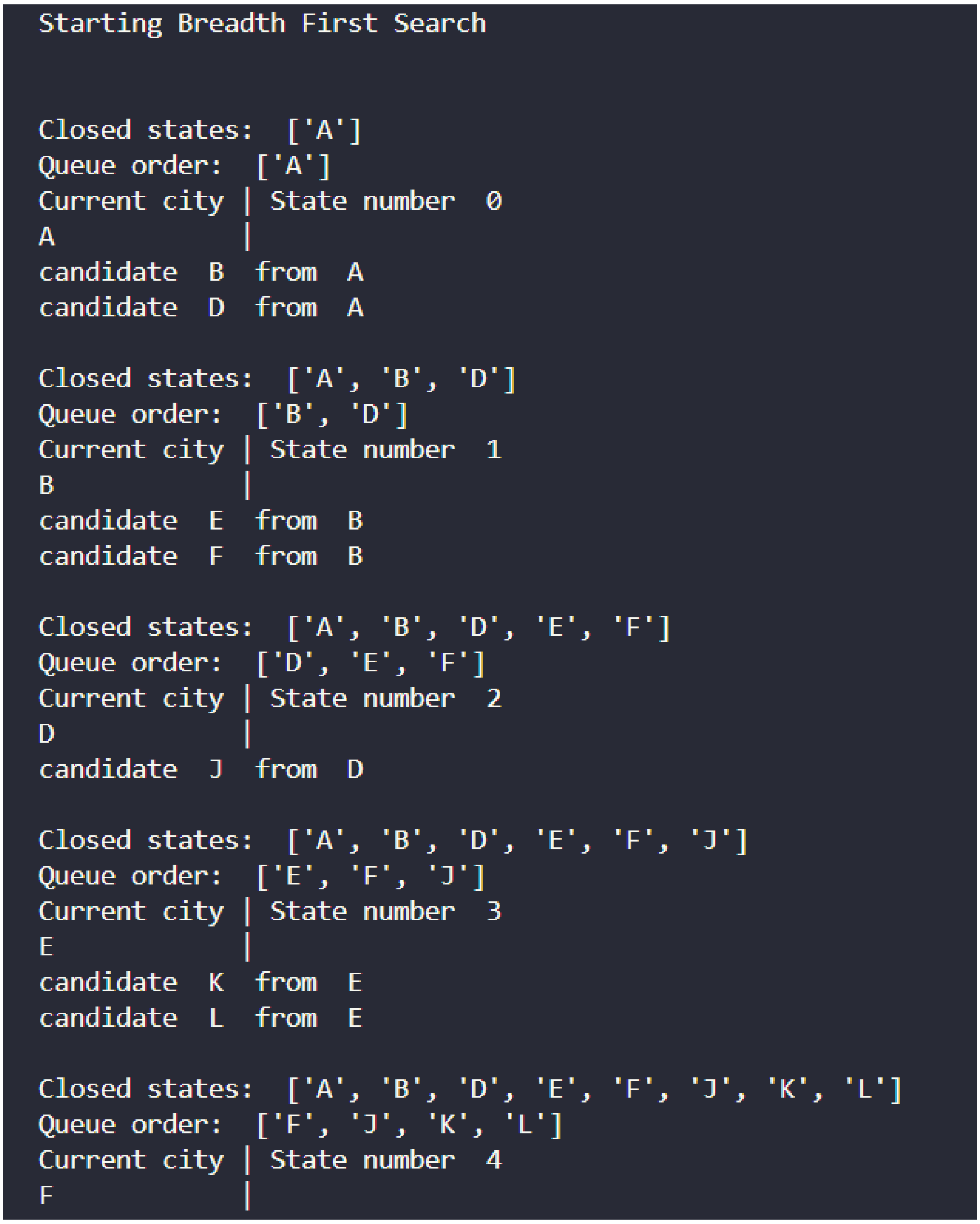
```
------Starting q3------

Closed states:  []
F | W | S | C | State number  0
L | L | L | L
Moving sheep from L to R


Closed states:  [('L', 'L', 'L', 'L')]
F | W | S | C | State number  1
R | L | R | L
Moving farmer from R to L


Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L')]
F | W | S | C | State number  2
L | L | R | L
Moving wolf from L to R


Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L'), ('L', 'L', 'R', 'L')]
F | W | S | C | State number  3
R | R | R | L
Moving sheep from R to L


Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L'), ('L', 'L', 'R', 'L'), ('R', 'R', 'R', 'L')]
F | W | S | C | State number  4
L | R | L | L
Moving cabbage from L to R
```



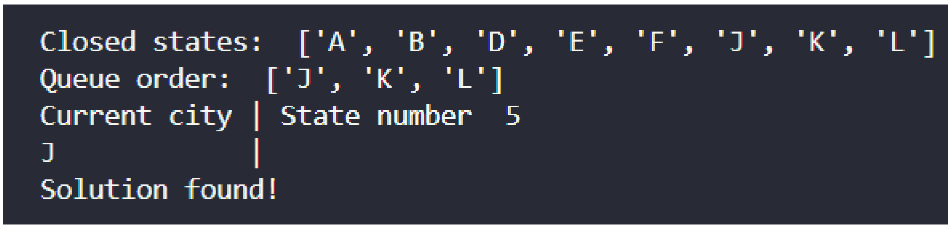
```
Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L'), ('L', 'L', 'R', 'L'), ('R', 'R', 'R', 'L'), ('L', 'R', 'L', 'L'
)]
F | W | S | C | State number  5
R | R | L | R
Moving farmer from R to L


Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L'), ('L', 'L', 'R', 'L'), ('R', 'R', 'R', 'L'), ('L', 'R', 'L', 'L'
), ('R', 'R', 'L', 'R')]
F | W | S | C | State number  6
L | R | L | R
Moving sheep from L to R


Closed states:  [('L', 'L', 'L', 'L'), ('R', 'L', 'R', 'L'), ('L', 'L', 'R', 'L'), ('R', 'R', 'R', 'L'), ('L', 'R', 'L', 'L'
), ('R', 'R', 'L', 'R'), ('L', 'R', 'L', 'R')]
F | W | S | C | State number  7
R | R | R | R
Solution found!

------Finishing q3------
```


Possible
Error
Impossible
Locked
Accepted
(L,L,L,L)
(R,L,L,L)
(R,R,L,L)
(R,L,R,L)
(R,L,L,R)
(L,L,R,L)
(L,R,R,L)
(L,L,L,L)
(L,L,L,R)
(R,L,R,L)
(R,R,R,L)
(R,L,L,L)
(R,L,R,R)
(L,R,R,L)
(L,L,R,L)
(L,R,L,L)
(L,R,R,R)
(R,R,L,L)
(R,L,L,L)
(R,R,R,L)
(R,R,L,R)
(L,R,L,R)
(L,L,L,R)
(L,R,R,R)
(L,R,L,L)
(R,R,L,R)
(R,L,L,R)
(R,R,R,R)
(R,R,L,L)

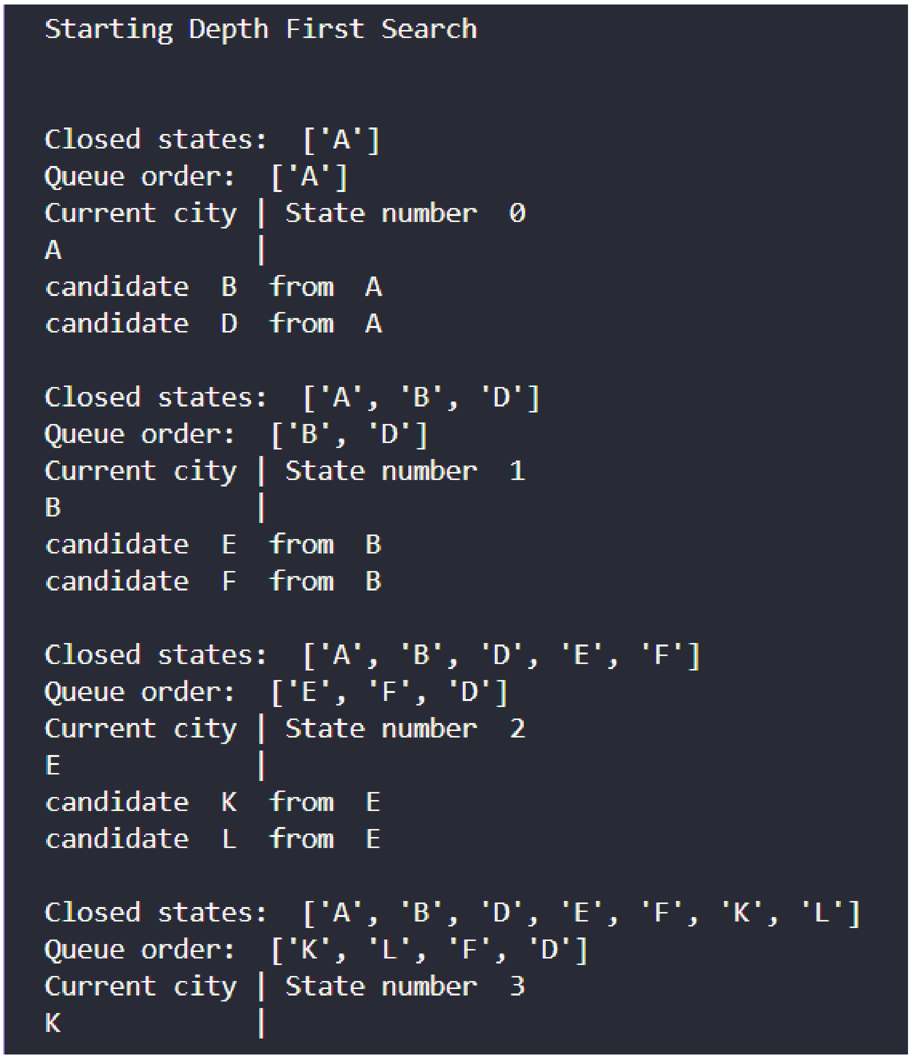
Questão 4:



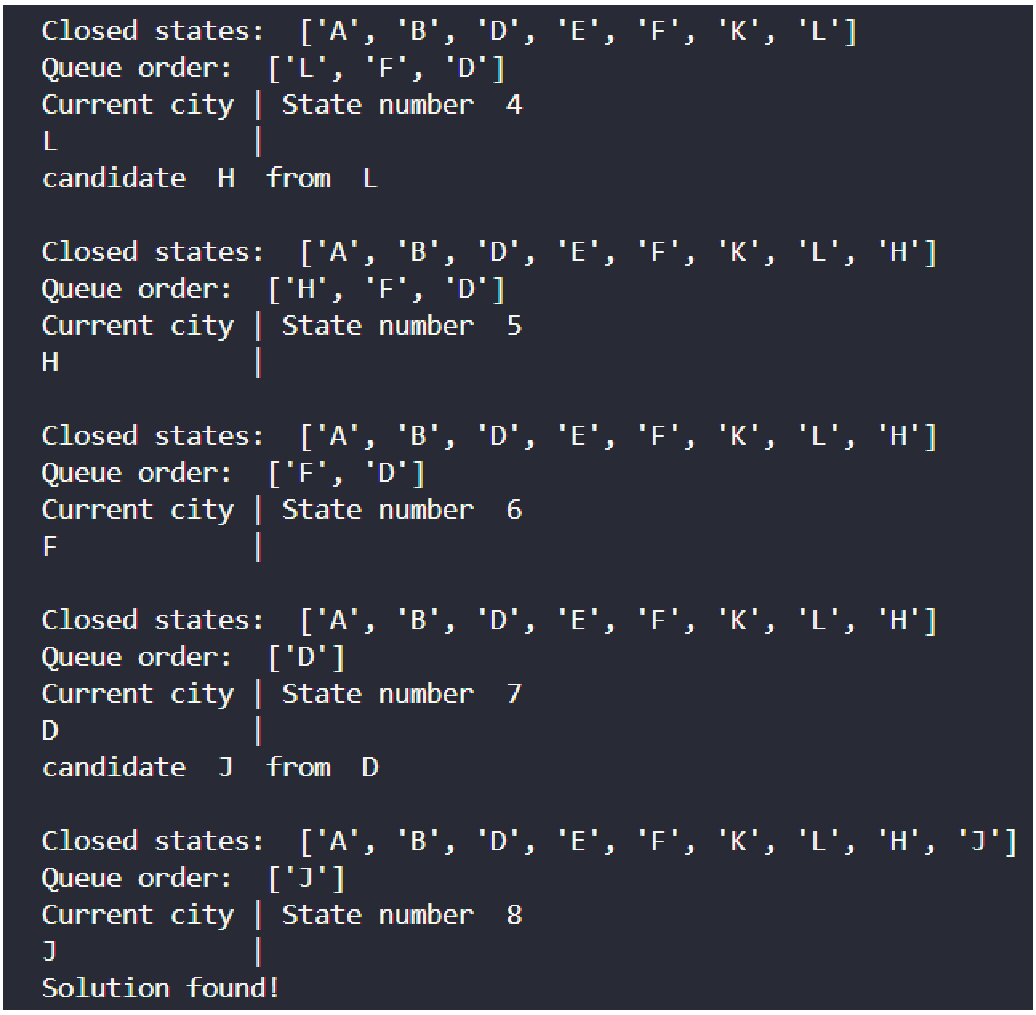
```
Starting Breadth First Search


Closed states:  ['A']
Queue order:  ['A']
Current city | State number  0
A            |
candidate  B  from  A
candidate  D  from  A

Closed states:  ['A', 'B', 'D']
Queue order:  ['B', 'D']
Current city | State number  1
B            |
candidate  E  from  B
candidate  F  from  B

Closed states:  ['A', 'B', 'D', 'E', 'F']
Queue order:  ['D', 'E', 'F']
Current city | State number  2
D            |
candidate  J  from  D

Closed states:  ['A', 'B', 'D', 'E', 'F', 'J']
Queue order:  ['E', 'F', 'J']
Current city | State number  3
E            |
candidate  K  from  E
candidate  L  from  E

Closed states:  ['A', 'B', 'D', 'E', 'F', 'J', 'K', 'L']
Queue order:  ['F', 'J', 'K', 'L']
Current city | State number  4
F            |
```



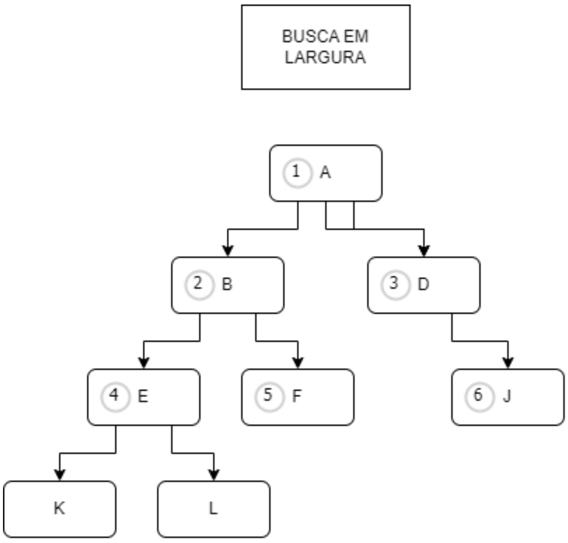
```
Closed states:  ['A', 'B', 'D', 'E', 'F', 'J', 'K', 'L']
Queue order:  ['J', 'K', 'L']
Current city | State number  5
J            |
Solution found!
```

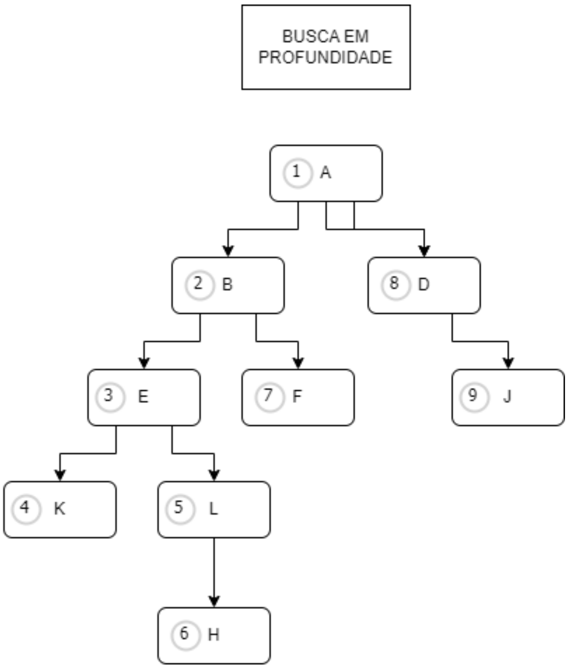


```
Starting Depth First Search


Closed states:  ['A']
Queue order:  ['A']
Current city | State number  0
A            |
candidate  B  from  A
candidate  D  from  A

Closed states:  ['A', 'B', 'D']
Queue order:  ['B', 'D']
Current city | State number  1
B            |
candidate  E  from  B
candidate  F  from  B

Closed states:  ['A', 'B', 'D', 'E', 'F']
Queue order:  ['E', 'F', 'D']
Current city | State number  2
E            |
candidate  K  from  E
candidate  L  from  E

Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L']
Queue order:  ['K', 'L', 'F', 'D']
Current city | State number  3
K            |
```

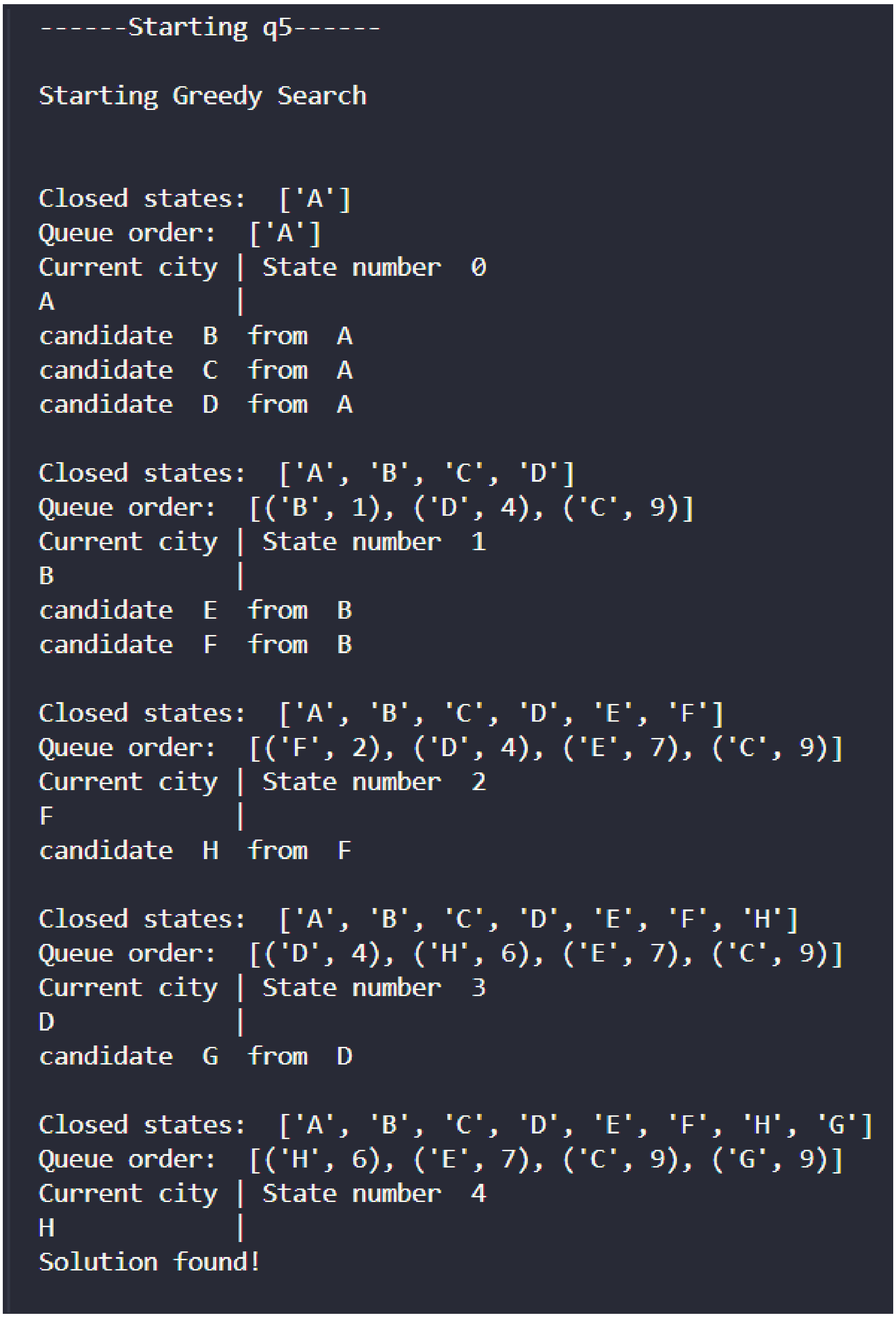


```
Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L']
Queue order:  ['L', 'F', 'D']
Current city | State number  4
L            |
candidate  H  from  L

Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L', 'H']
Queue order:  ['H', 'F', 'D']
Current city | State number  5
H            |

Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L', 'H']
Queue order:  ['F', 'D']
Current city | State number  6
F            |

Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L', 'H']
Queue order:  ['D']
Current city | State number  7
D            |
candidate  J  from  D

Closed states:  ['A', 'B', 'D', 'E', 'F', 'K', 'L', 'H', 'J']
Queue order:  ['J']
Current city | State number  8
J            |
Solution found!
```

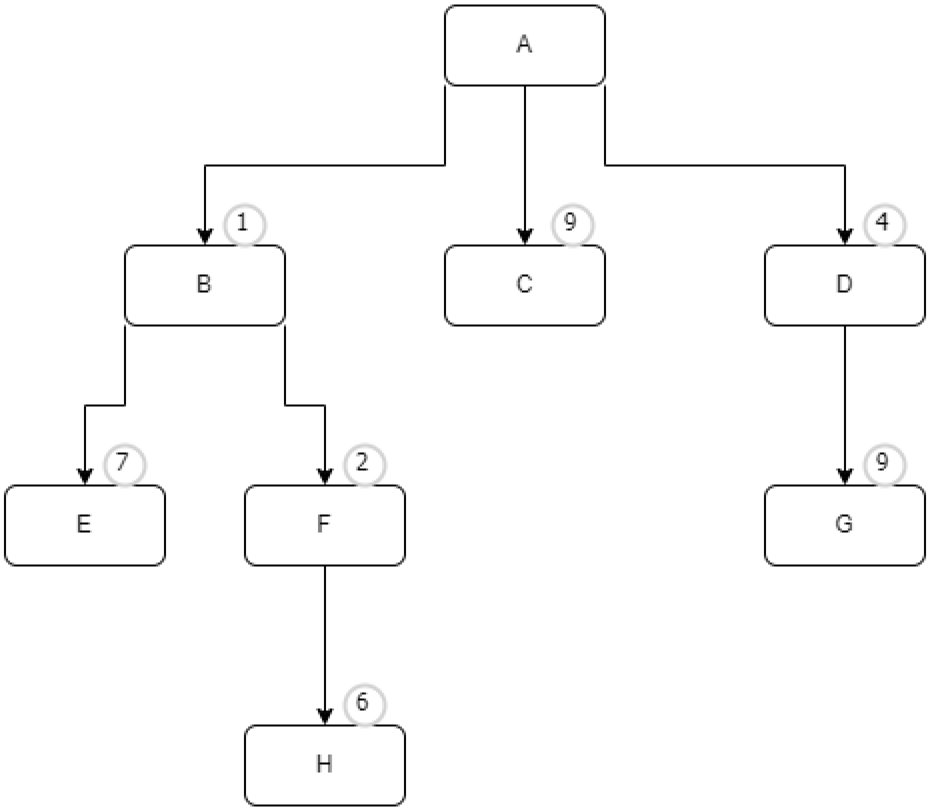




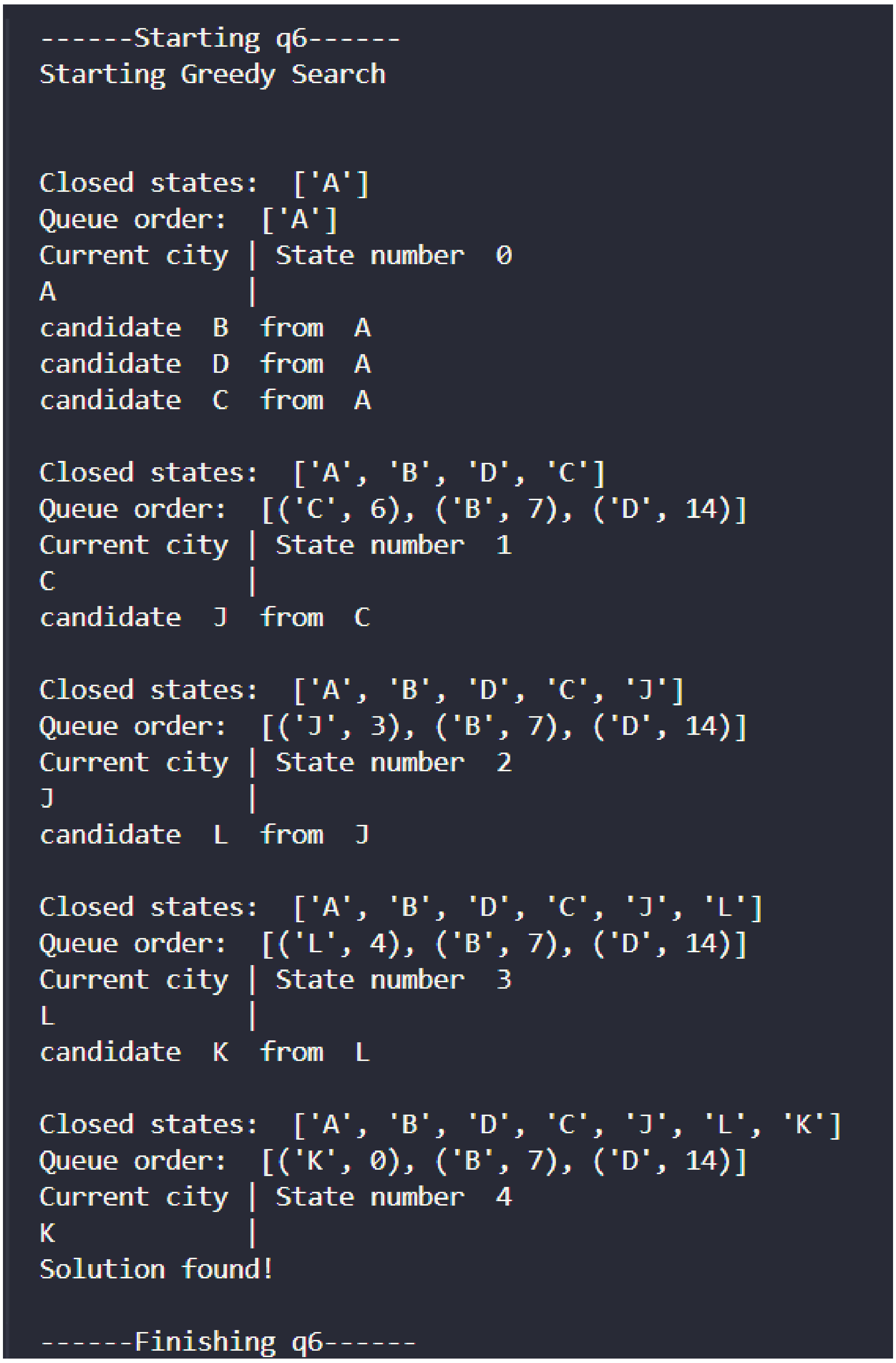
Questão 5:



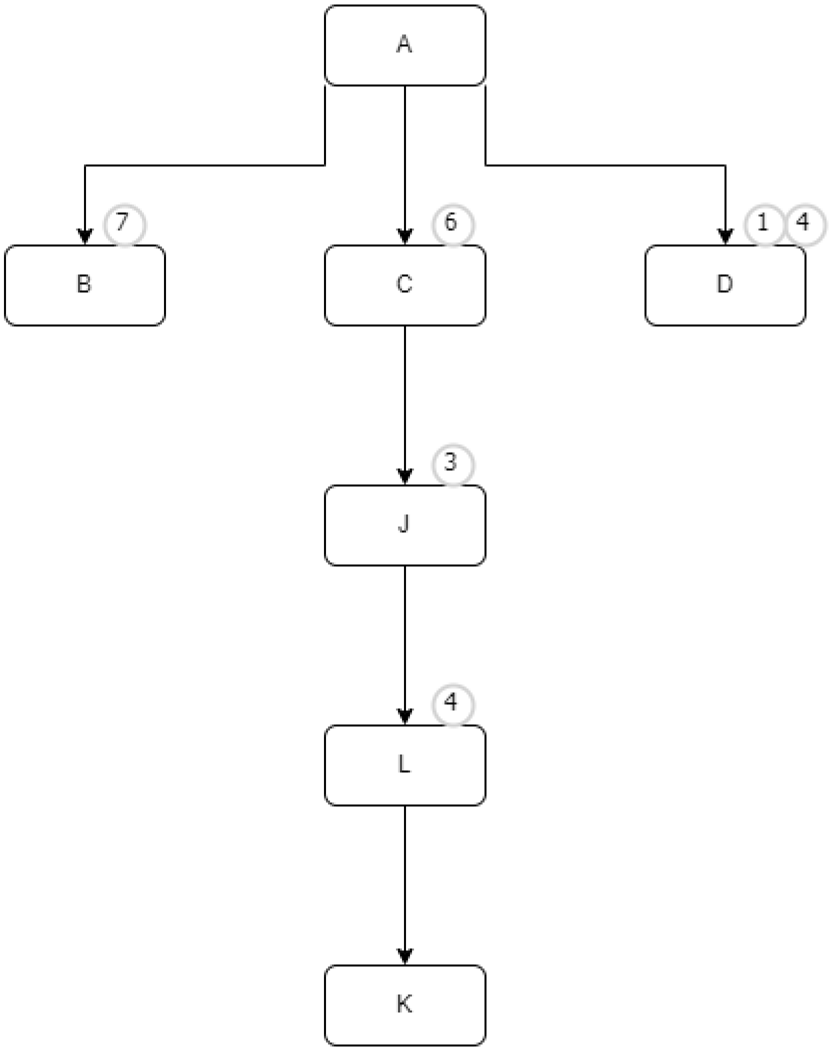
```
------Starting q5------

Starting Greedy Search


Closed states:  ['A']
Queue order:  ['A']
Current city | State number  0
A            |
candidate  B  from  A
candidate  C  from  A
candidate  D  from  A

Closed states:  ['A', 'B', 'C', 'D']
Queue order:  [('B', 1), ('D', 4), ('C', 9)]
Current city | State number  1
B            |
candidate  E  from  B
candidate  F  from  B

Closed states:  ['A', 'B', 'C', 'D', 'E', 'F']
Queue order:  [('F', 2), ('D', 4), ('E', 7), ('C', 9)]
Current city | State number  2
F            |
candidate  H  from  F

Closed states:  ['A', 'B', 'C', 'D', 'E', 'F', 'H']
Queue order:  [('D', 4), ('H', 6), ('E', 7), ('C', 9)]
Current city | State number  3
D            |
candidate  G  from  D

Closed states:  ['A', 'B', 'C', 'D', 'E', 'F', 'H', 'G']
Queue order:  [('H', 6), ('E', 7), ('C', 9), ('G', 9)]
Current city | State number  4
H            |
Solution found!
```


A
1
B
9
C
4
D
7
E
2
F
9
G
6
H

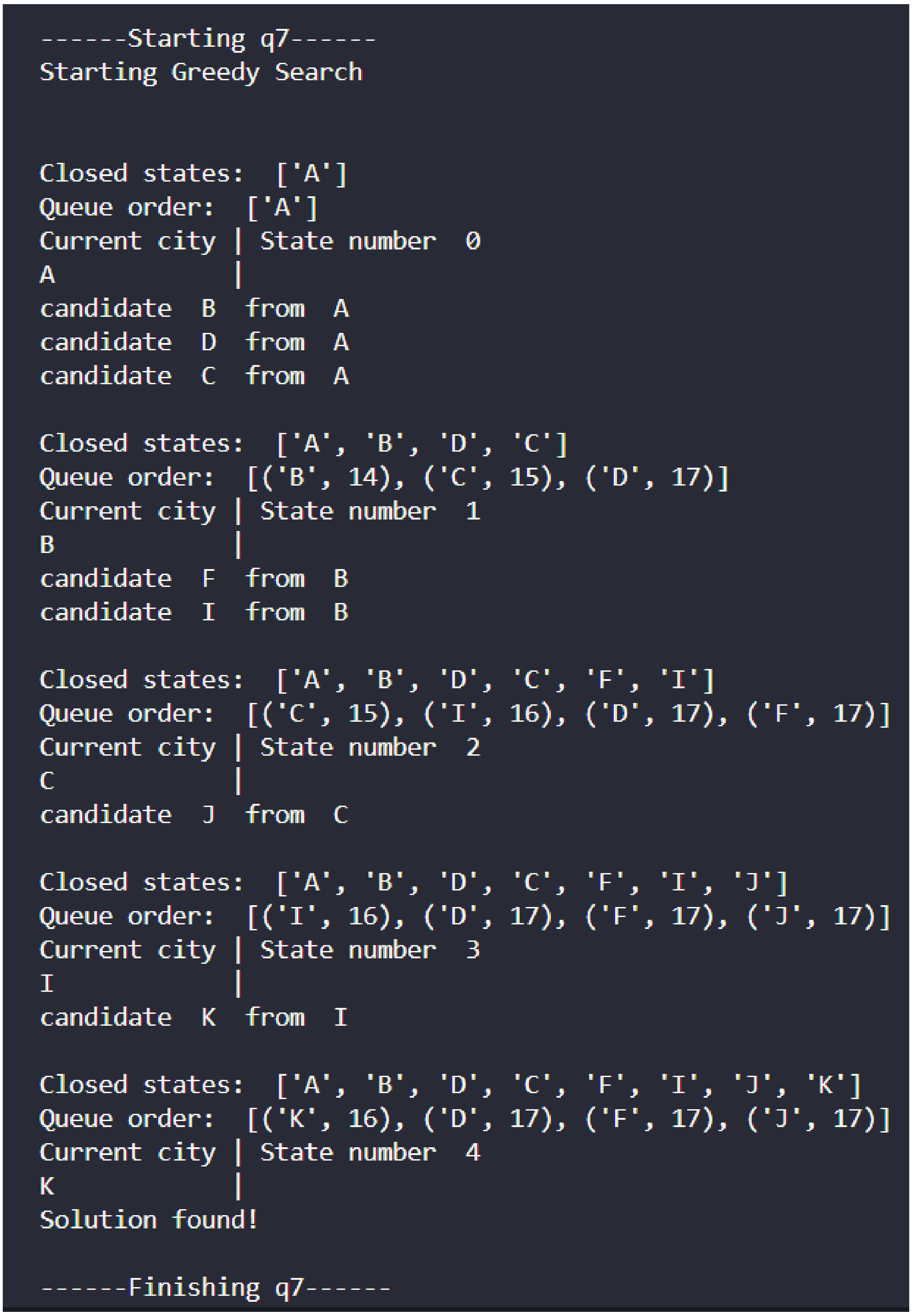
Questão 6:



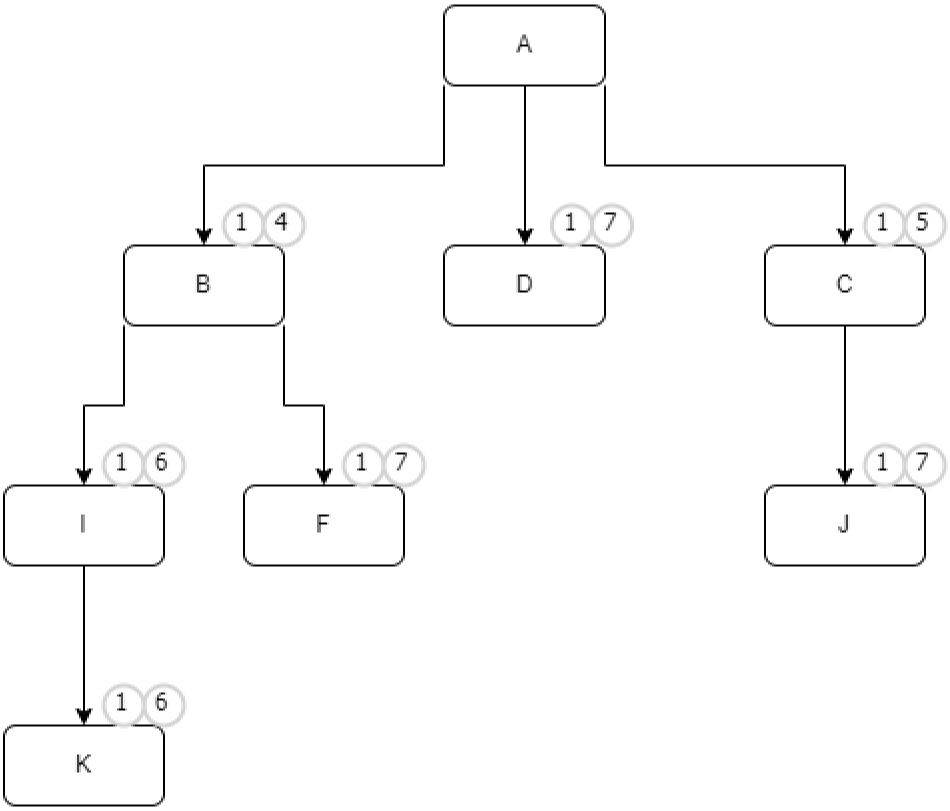
```
------Starting q6------
Starting Greedy Search


Closed states:  ['A']
Queue order:  ['A']
Current city | State number  0
A            |
candidate  B  from  A
candidate  D  from  A
candidate  C  from  A

Closed states:  ['A', 'B', 'D', 'C']
Queue order:  [('C', 6), ('B', 7), ('D', 14)]
Current city | State number  1
C            |
candidate  J  from  C

Closed states:  ['A', 'B', 'D', 'C', 'J']
Queue order:  [('J', 3), ('B', 7), ('D', 14)]
Current city | State number  2
J            |
candidate  L  from  J

Closed states:  ['A', 'B', 'D', 'C', 'J', 'L']
Queue order:  [('L', 4), ('B', 7), ('D', 14)]
Current city | State number  3
L            |
candidate  K  from  L

Closed states:  ['A', 'B', 'D', 'C', 'J', 'L', 'K']
Queue order:  [('K', 0), ('B', 7), ('D', 14)]
Current city | State number  4
K            |
Solution found!

------Finishing q6------
```



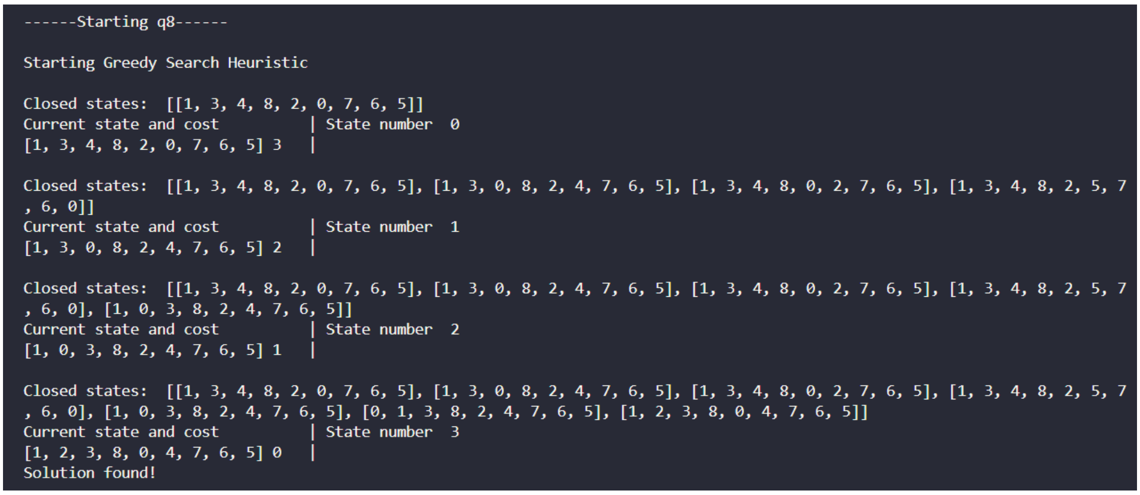
Questão 7:



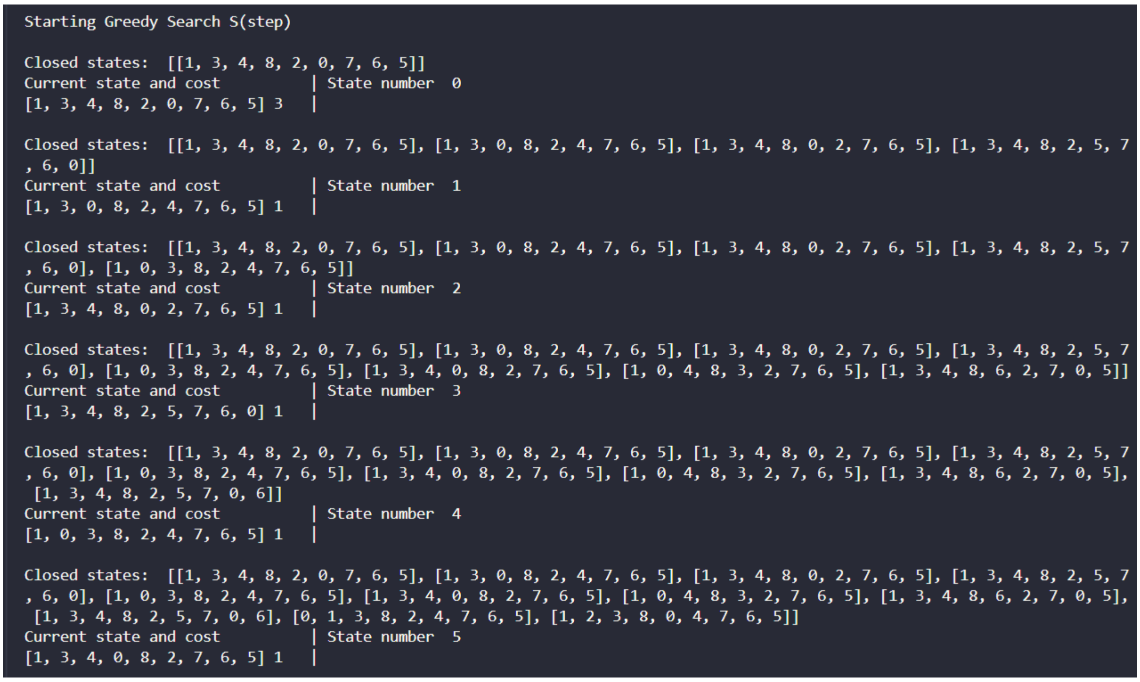
```
------Starting q7------
Starting Greedy Search


Closed states:  ['A']
Queue order:  ['A']
Current city | State number  0
A            |
candidate  B  from  A
candidate  D  from  A
candidate  C  from  A

Closed states:  ['A', 'B', 'D', 'C']
Queue order:  [('B', 14), ('C', 15), ('D', 17)]
Current city | State number  1
B            |
candidate  F  from  B
candidate  I  from  B

Closed states:  ['A', 'B', 'D', 'C', 'F', 'I']
Queue order:  [('C', 15), ('I', 16), ('D', 17), ('F', 17)]
Current city | State number  2
C            |
candidate  J  from  C

Closed states:  ['A', 'B', 'D', 'C', 'F', 'I', 'J']
Queue order:  [('I', 16), ('D', 17), ('F', 17), ('J', 17)]
Current city | State number  3
I            |
candidate  K  from  I

Closed states:  ['A', 'B', 'D', 'C', 'F', 'I', 'J', 'K']
Queue order:  [('K', 16), ('D', 17), ('F', 17), ('J', 17)]
Current city | State number  4
K            |
Solution found!

------Finishing q7------
```


A
1
4
B
1
7
D
1
5
C
1
6
I
1
7
F
1
7
J
1
6
K

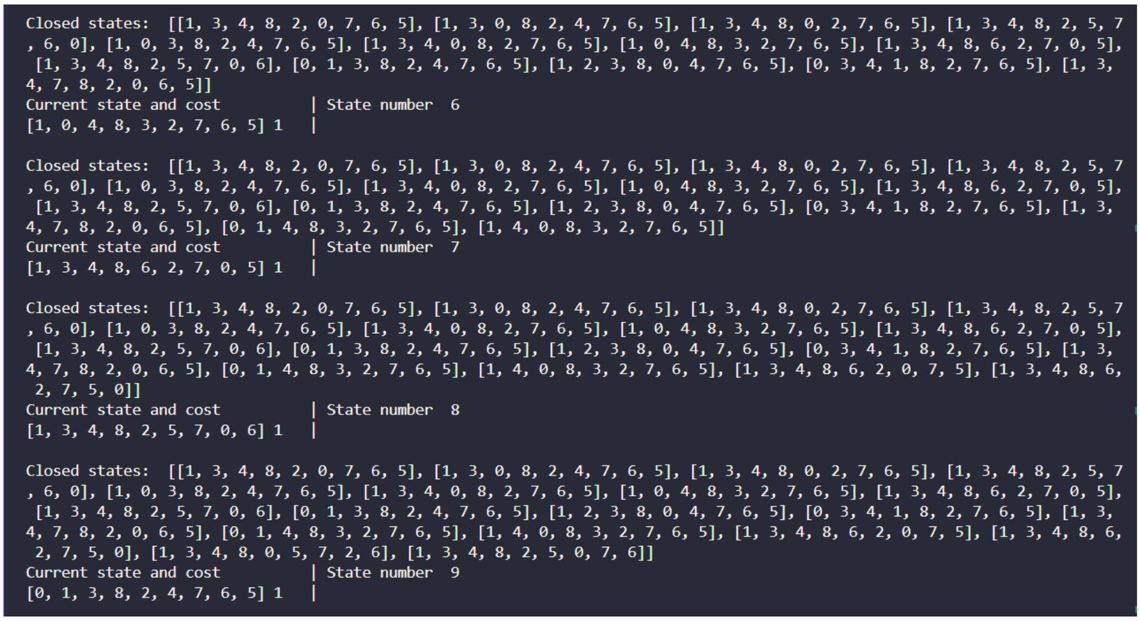
Questão 8:



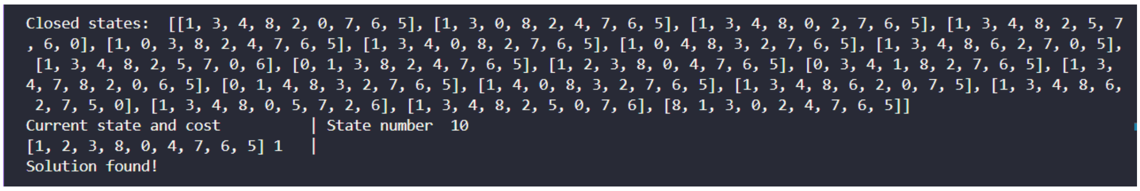
```
------Starting q8------

Starting Greedy Search Heuristic

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5]]
Current state and cost          | State number  0
[1, 3, 4, 8, 2, 0, 7, 6, 5] 3   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0]]
Current state and cost          | State number  1
[1, 3, 0, 8, 2, 4, 7, 6, 5] 2   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5]]
Current state and cost          | State number  2
[1, 0, 3, 8, 2, 4, 7, 6, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5]]
Current state and cost          | State number  3
[1, 2, 3, 8, 0, 4, 7, 6, 5] 0   |
Solution found!
```



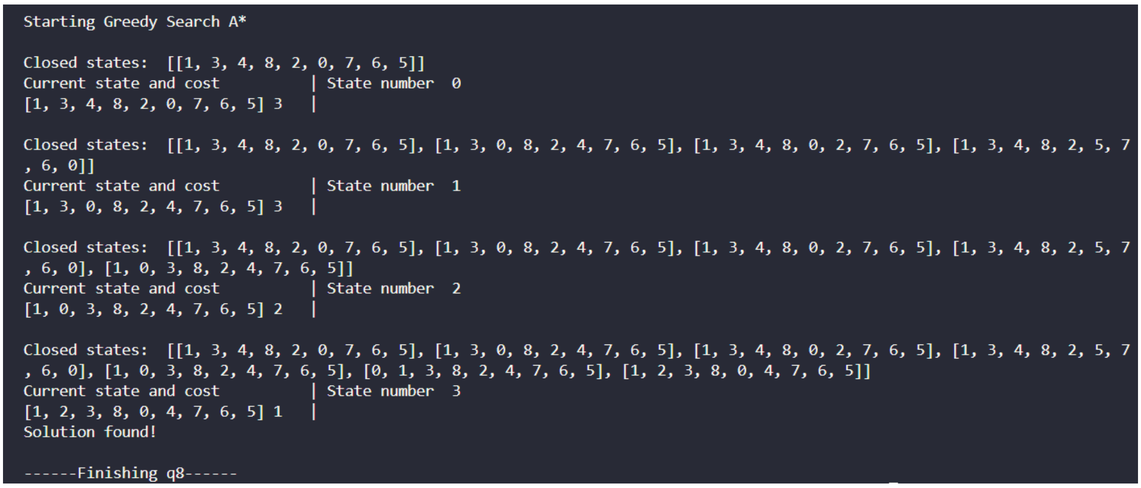
```
Starting Greedy Search S(step)

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5]]
Current state and cost          | State number  0
[1, 3, 4, 8, 2, 0, 7, 6, 5] 3   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0]]
Current state and cost          | State number  1
[1, 3, 0, 8, 2, 4, 7, 6, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5]]
Current state and cost          | State number  2
[1, 3, 4, 8, 0, 2, 7, 6, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5]]
Current state and cost          | State number  3
[1, 3, 4, 8, 2, 5, 7, 6, 0] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6]]
Current state and cost          | State number  4
[1, 0, 3, 8, 2, 4, 7, 6, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5]]
Current state and cost          | State number  5
[1, 3, 4, 0, 8, 2, 7, 6, 5] 1   |
```



```
Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5], [0, 3, 4, 1, 8, 2, 7, 6, 5], [1, 3,
4, 7, 8, 2, 0, 6, 5]]
Current state and cost          | State number  6
[1, 0, 4, 8, 3, 2, 7, 6, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5], [0, 3, 4, 1, 8, 2, 7, 6, 5], [1, 3,
4, 7, 8, 2, 0, 6, 5], [0, 1, 4, 8, 3, 2, 7, 6, 5], [1, 4, 0, 8, 3, 2, 7, 6, 5]]
Current state and cost          | State number  7
[1, 3, 4, 8, 6, 2, 7, 0, 5] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5], [0, 3, 4, 1, 8, 2, 7, 6, 5], [1, 3,
4, 7, 8, 2, 0, 6, 5], [0, 1, 4, 8, 3, 2, 7, 6, 5], [1, 4, 0, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 0, 7, 5], [1, 3, 4, 8, 6,
 2, 7, 5, 0]]
Current state and cost          | State number  8
[1, 3, 4, 8, 2, 5, 7, 0, 6] 1   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5], [0, 3, 4, 1, 8, 2, 7, 6, 5], [1, 3,
4, 7, 8, 2, 0, 6, 5], [0, 1, 4, 8, 3, 2, 7, 6, 5], [1, 4, 0, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 0, 7, 5], [1, 3, 4, 8, 6,
 2, 7, 5, 0], [1, 3, 4, 8, 0, 5, 7, 2, 6], [1, 3, 4, 8, 2, 5, 0, 7, 6]]
Current state and cost          | State number  9
[0, 1, 3, 8, 2, 4, 7, 6, 5] 1   |
```

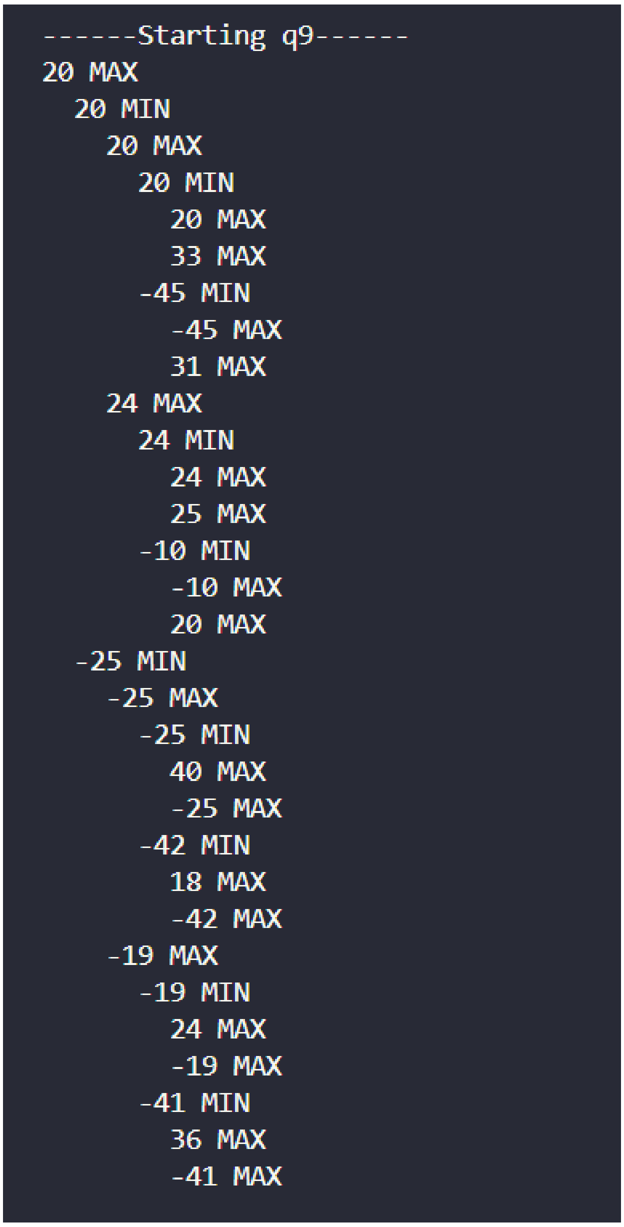


```
Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [1, 3, 4, 0, 8, 2, 7, 6, 5], [1, 0, 4, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 7, 0, 5],
 [1, 3, 4, 8, 2, 5, 7, 0, 6], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5], [0, 3, 4, 1, 8, 2, 7, 6, 5], [1, 3,
4, 7, 8, 2, 0, 6, 5], [0, 1, 4, 8, 3, 2, 7, 6, 5], [1, 4, 0, 8, 3, 2, 7, 6, 5], [1, 3, 4, 8, 6, 2, 0, 7, 5], [1, 3, 4, 8, 6,
 2, 7, 5, 0], [1, 3, 4, 8, 0, 5, 7, 2, 6], [1, 3, 4, 8, 2, 5, 0, 7, 6], [8, 1, 3, 0, 2, 4, 7, 6, 5]]
Current state and cost          | State number  10
[1, 2, 3, 8, 0, 4, 7, 6, 5] 1   |
Solution found!
```

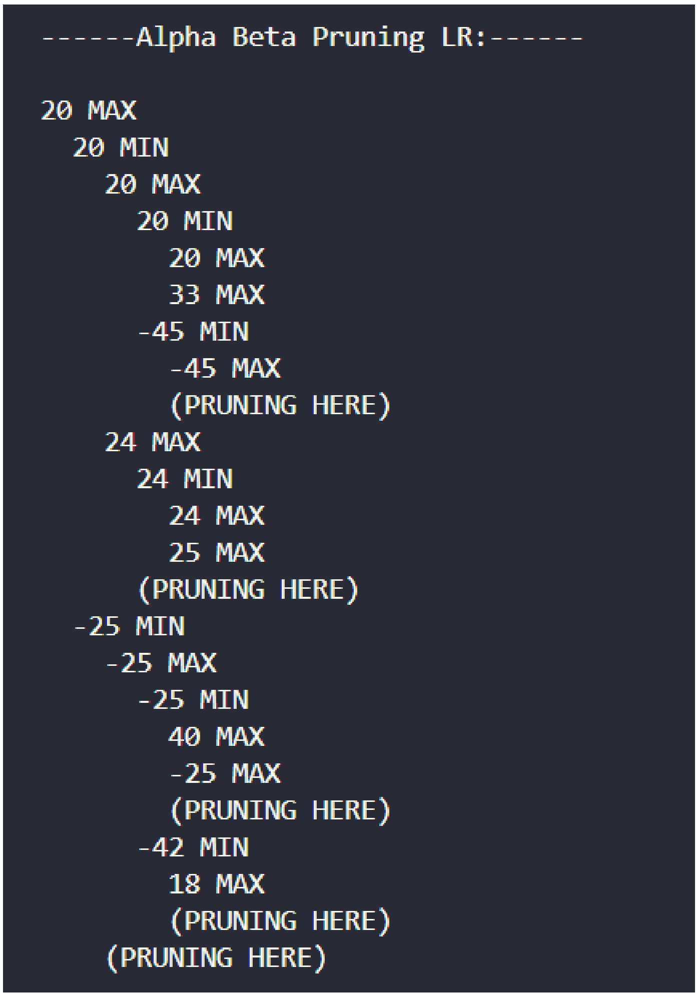


```
Starting Greedy Search A*

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5]]
Current state and cost          | State number  0
[1, 3, 4, 8, 2, 0, 7, 6, 5] 3   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0]]
Current state and cost          | State number  1
[1, 3, 0, 8, 2, 4, 7, 6, 5] 3   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5]]
Current state and cost          | State number  2
[1, 0, 3, 8, 2, 4, 7, 6, 5] 2   |

Closed states:  [[1, 3, 4, 8, 2, 0, 7, 6, 5], [1, 3, 0, 8, 2, 4, 7, 6, 5], [1, 3, 4, 8, 0, 2, 7, 6, 5], [1, 3, 4, 8, 2, 5, 7
, 6, 0], [1, 0, 3, 8, 2, 4, 7, 6, 5], [0, 1, 3, 8, 2, 4, 7, 6, 5], [1, 2, 3, 8, 0, 4, 7, 6, 5]]
Current state and cost          | State number  3
[1, 2, 3, 8, 0, 4, 7, 6, 5] 1   |
Solution found!

------Finishing q8------
```

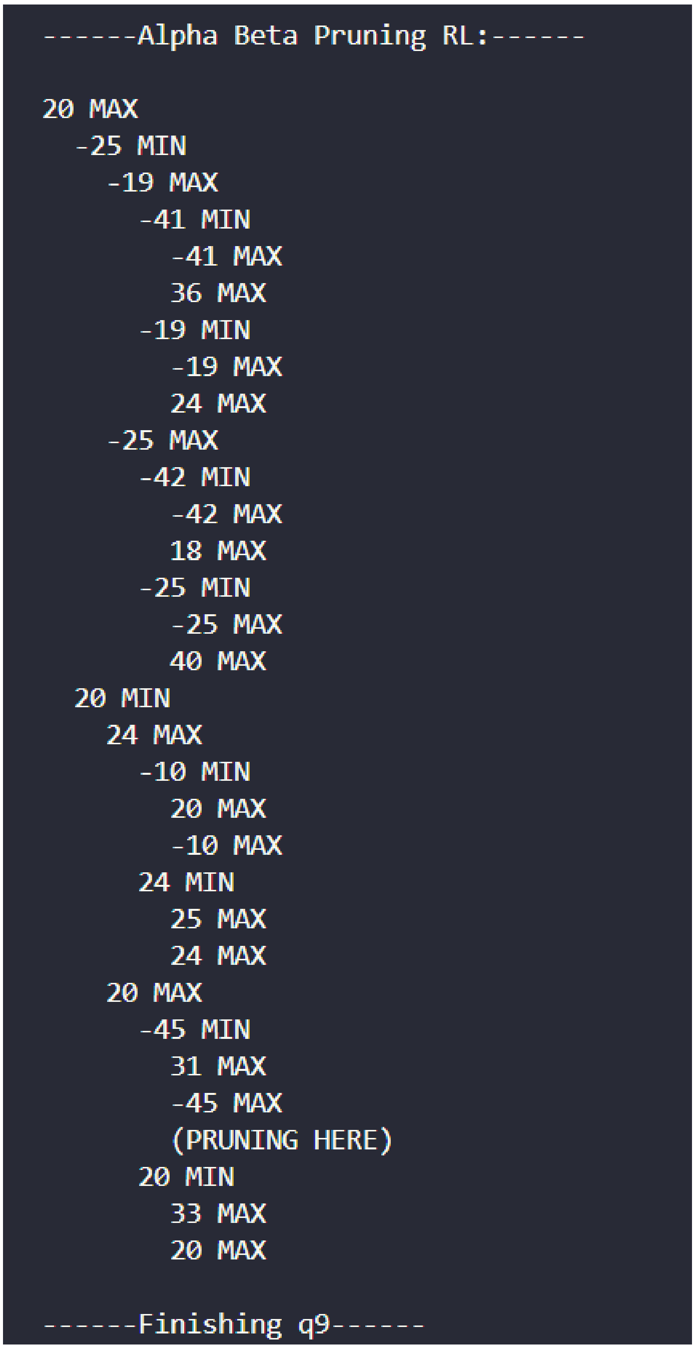
Questão 9:



```
------Starting q9------
20 MAX
  20 MIN
    20 MAX
      20 MIN
        20 MAX
        33 MAX
      -45 MIN
        -45 MAX
        31 MAX
    24 MAX
      24 MIN
        24 MAX
        25 MAX
      -10 MIN
        -10 MAX
        20 MAX
  -25 MIN
    -25 MAX
      -25 MIN
        40 MAX
        -25 MAX
      -42 MIN
        18 MAX
        -42 MAX
    -19 MAX
      -19 MIN
        24 MAX
        -19 MAX
      -41 MIN
        36 MAX
        -41 MAX
```

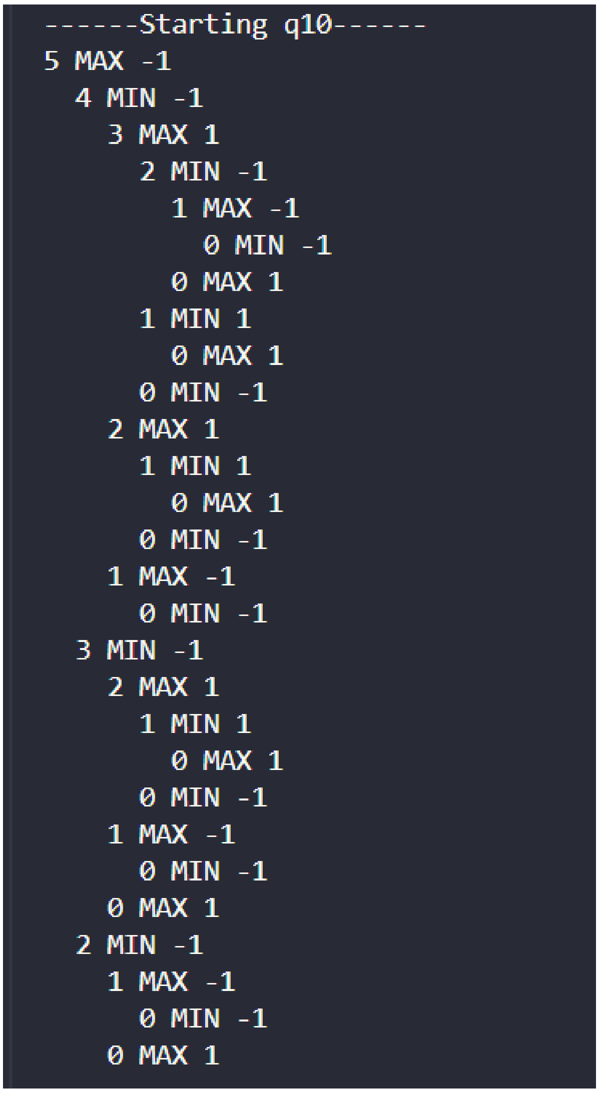


```
------Alpha Beta Pruning LR:------

20 MAX
  20 MIN
    20 MAX
      20 MIN
        20 MAX
        33 MAX
      -45 MIN
        -45 MAX
        (PRUNING HERE)
    24 MAX
      24 MIN
        24 MAX
        25 MAX
      (PRUNING HERE)
  -25 MIN
    -25 MAX
      -25 MIN
        40 MAX
        -25 MAX
        (PRUNING HERE)
      -42 MIN
        18 MAX
        (PRUNING HERE)
    (PRUNING HERE)
```

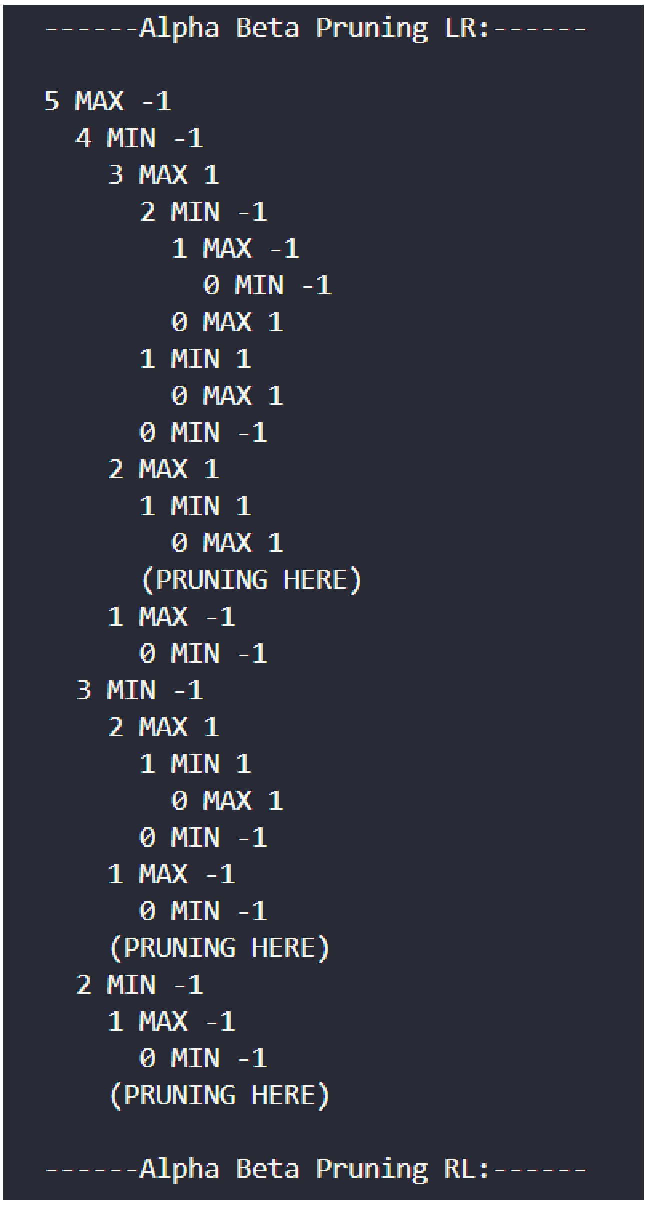


```
------Alpha Beta Pruning RL:------

20 MAX
  -25 MIN
    -19 MAX
      -41 MIN
        -41 MAX
        36 MAX
      -19 MIN
        -19 MAX
        24 MAX
    -25 MAX
      -42 MIN
        -42 MAX
        18 MAX
      -25 MIN
        -25 MAX
        40 MAX
  20 MIN
    24 MAX
      -10 MIN
        20 MAX
        -10 MAX
      24 MIN
        25 MAX
        24 MAX
    20 MAX
      -45 MIN
        31 MAX
        -45 MAX
        (PRUNING HERE)
      20 MIN
        33 MAX
        20 MAX

------Finishing q9------
```

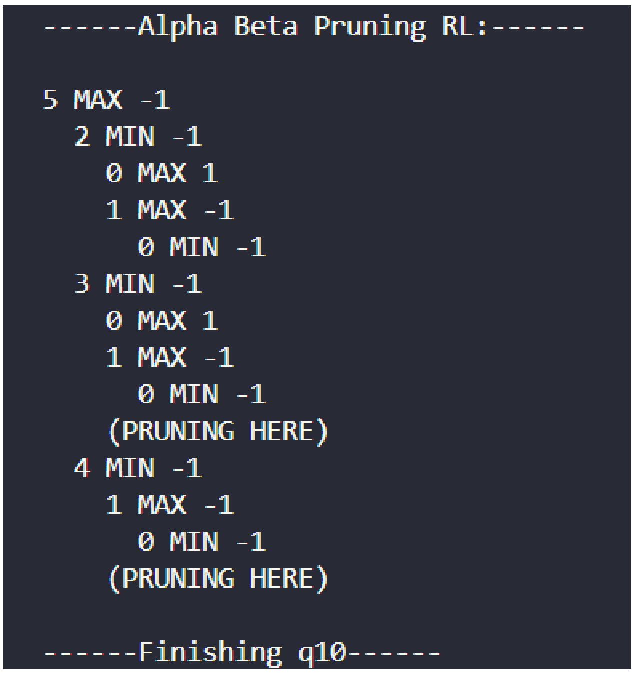
Questão 10:



```
------Starting q10------
5 MAX -1
  4 MIN -1
    3 MAX 1
      2 MIN -1
        1 MAX -1
          0 MIN -1
        0 MAX 1
      1 MIN 1
        0 MAX 1
      0 MIN -1
    2 MAX 1
      1 MIN 1
        0 MAX 1
      0 MIN -1
    1 MAX -1
      0 MIN -1
  3 MIN -1
    2 MAX 1
      1 MIN 1
        0 MAX 1
      0 MIN -1
    1 MAX -1
      0 MIN -1
    0 MAX 1
  2 MIN -1
    1 MAX -1
      0 MIN -1
    0 MAX 1
```



```
------Alpha Beta Pruning LR:------

5 MAX -1
  4 MIN -1
    3 MAX 1
      2 MIN -1
        1 MAX -1
          0 MIN -1
        0 MAX 1
      1 MIN 1
        0 MAX 1
      0 MIN -1
    2 MAX 1
      1 MIN 1
        0 MAX 1
      (PRUNING HERE)
    1 MAX -1
      0 MIN -1
  3 MIN -1
    2 MAX 1
      1 MIN 1
        0 MAX 1
      0 MIN -1
    1 MAX -1
      0 MIN -1
    (PRUNING HERE)
  2 MIN -1
    1 MAX -1
      0 MIN -1
    (PRUNING HERE)

------Alpha Beta Pruning RL:------
```



```
------Alpha Beta Pruning RL:------

5 MAX -1
  2 MIN -1
    0 MAX 1
    1 MAX -1
      0 MIN -1
  3 MIN -1
    0 MAX 1
    1 MAX -1
      0 MIN -1
    (PRUNING HERE)
  4 MIN -1
    1 MAX -1
      0 MIN -1
    (PRUNING HERE)

------Finishing q10------
```