

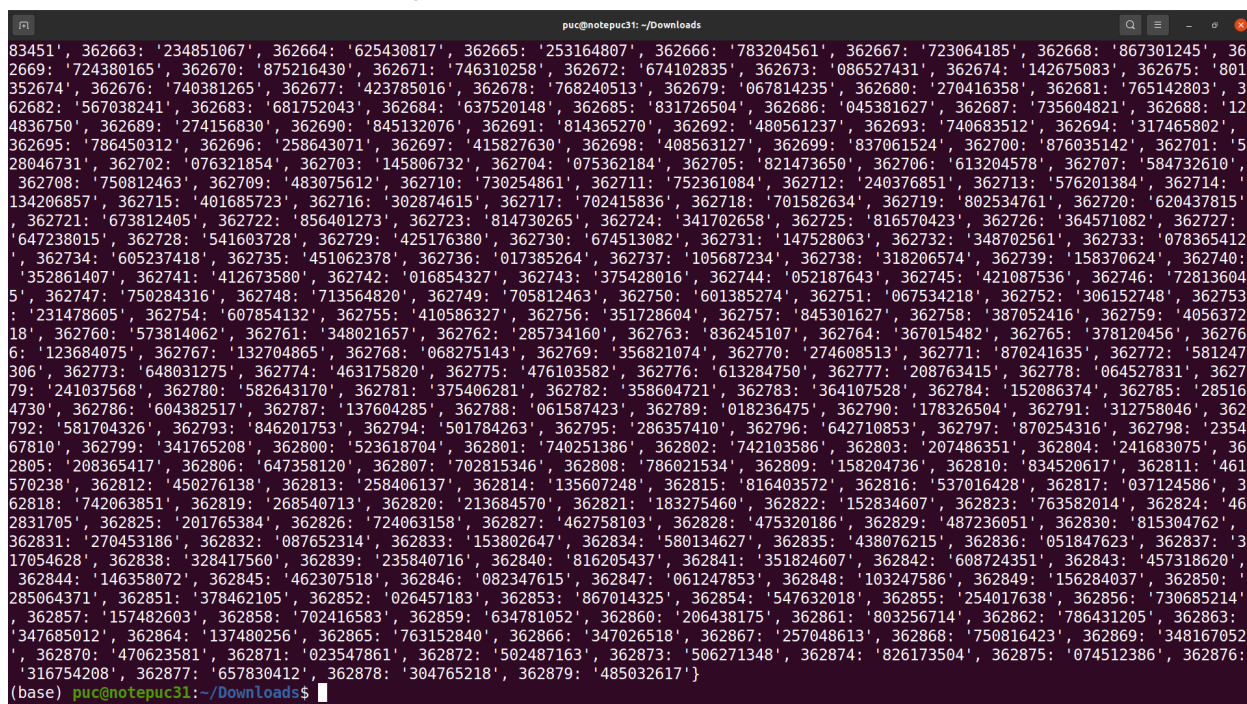
T2 - Análise de Algoritmos

Jam Ajna Soares

Gabriel Andrade

Foram criadas algumas funções auxiliares e também de acordo com as dicas do enunciado, porém nem todas foram utilizadas (por exemplo o vetor C) e algumas outras foram criadas por se fazerem necessárias.

De início após a confirmação do funcionamento do algoritmo tentamos preencher a Tabela Hash H de 9! (362880) nós - um para cada estado possível do tabuleiro - com uma random seed. O que com menos caracteres pareceu elegante se transformou em mais de 4 milhões de iterações até completar a tabela, e quando isso enfim aconteceu um print da Tabela completa sumiu com toda a informação que seria mostrada na tela respondendo as perguntas do relatório.



```
puc@notepuc31:~/Downloads
83451', 362663: '234851067', 362664: '625430817', 362665: '253164807', 362666: '783204561', 362667: '723064185', 362668: '867301245', 36
2669: '724380165', 362670: '875216430', 362671: '746310258', 362672: '674102835', 362673: '086527431', 362674: '142675083', 362675: '801
352674', 362676: '740381265', 362677: '423785016', 362678: '768240513', 362679: '067814235', 362680: '270416358', 362681: '765142803', 3
62682: '567038241', 362683: '681752043', 362684: '637520148', 362685: '831726504', 362686: '045381627', 362687: '735604821', 362688: '12
4836750', 362689: '274156830', 362690: '845132076', 362691: '814365270', 362692: '480561237', 362693: '740683512', 362694: '317465802',
362695: '786450312', 362696: '258643071', 362697: '415827630', 362698: '408563127', 362699: '837061524', 362700: '876035142', 362701: '5
28046731', 362702: '076321854', 362703: '145806732', 362704: '075362184', 362705: '821473650', 362706: '613204578', 362707: '584732610',
362708: '750812463', 362709: '483075612', 362710: '730254861', 362711: '752361084', 362712: '240376851', 362713: '576201384', 362714: '1
134206857', 362715: '401685723', 362716: '302874615', 362717: '702415836', 362718: '701582634', 362719: '802534761', 362720: '620437815',
362721: '673812405', 362722: '856401273', 362723: '814730265', 362724: '341702658', 362725: '816570423', 362726: '364571082', 362727:
'647238015', 362728: '541603728', 362729: '425176380', 362730: '674513082', 362731: '147528063', 362732: '348702561', 362733: '078365412',
362734: '605237418', 362735: '451062378', 362736: '017385264', 362737: '105687234', 362738: '318206574', 362739: '158370624', 362740: '3
352861407', 362741: '412673580', 362742: '016854327', 362743: '375428016', 362744: '052187643', 362745: '421087536', 362746: '72813604
5', 362747: '750284316', 362748: '713564820', 362749: '705812463', 362750: '601385274', 362751: '067534218', 362752: '306152748', 362753
: '231478605', 362754: '607854132', 362755: '410586327', 362756: '351728604', 362757: '845301627', 362758: '387052416', 362759: '4056372
18', 362760: '573814062', 362761: '348021657', 362762: '285734160', 362763: '836245107', 362764: '367015482', 362765: '378120456', 36276
6: '123684075', 362767: '132704865', 362768: '068275143', 362769: '356821074', 362770: '274608513', 362771: '870241635', 362772: '581247
306', 362773: '648031275', 362774: '463175820', 362775: '476103582', 362776: '613284750', 362777: '208763415', 362778: '064527831', 3627
79: '241037568', 362780: '582643170', 362781: '375406281', 362782: '358604721', 362783: '364107528', 362784: '152086374', 362785: '28516
4730', 362786: '604382517', 362787: '137604285', 362788: '061587423', 362789: '018236475', 362790: '178326504', 362791: '312758046', 362
792: '581704326', 362793: '846201753', 362794: '501784263', 362795: '286357410', 362796: '642710853', 362797: '870254316', 362798: '2354
67810', 362799: '341765208', 362800: '523618704', 362801: '740251386', 362802: '742103586', 362803: '207486351', 362804: '241683075', 36
2805: '208365417', 362806: '647358120', 362807: '702815346', 362808: '786021534', 362809: '158204736', 362810: '834520617', 362811: '461
570238', 362812: '450276138', 362813: '258406137', 362814: '135607248', 362815: '816403572', 362816: '537016428', 362817: '037124586', 3
62818: '742063851', 362819: '268540713', 362820: '213684570', 362821: '183275460', 362822: '152834607', 362823: '763582014', 362824: '46
2831705', 362825: '201765384', 362826: '724063158', 362827: '462758103', 362828: '475320186', 362829: '487236051', 362830: '815304762',
362831: '270453186', 362832: '087652314', 362833: '153802647', 362834: '580134627', 362835: '438076215', 362836: '051847623', 362837: '3
17054628', 362838: '328417560', 362839: '235840716', 362840: '816205437', 362841: '351824607', 362842: '608724351', 362843: '457318620',
362844: '146358072', 362845: '462307518', 362846: '082347615', 362847: '061247853', 362848: '103247586', 362849: '156284037', 362850: '2
285064371', 362851: '378462105', 362852: '026457183', 362853: '867014325', 362854: '547632018', 362855: '254017638', 362856: '730685214',
362857: '157482603', 362858: '702416583', 362859: '634781052', 362860: '206438175', 362861: '803256714', 362862: '786431205', 362863: '3
347685012', 362864: '137480256', 362865: '763152840', 362866: '347026518', 362867: '257048613', 362868: '750816423', 362869: '348167052',
362870: '470623581', 362871: '023547861', 362872: '502487163', 362873: '506271348', 362874: '826173504', 362875: '074512386', 362876: '3
316754208', 362877: '657830412', 362878: '304765218', 362879: '485032617'}
```

(base) puc@notepuc31:~/Downloads\$

Essa estratégia foi abandonada usando-se no lugar um módulo de permutação criando os nós em tempo linear:

```
pu@notepuc31: ~  
  
Nó 210724 adicionado à tabela, com configuração 731854260  
Nó 210726 adicionado à tabela, com configuração 731854620  
Nó 210728 adicionado à tabela, com configuração 731850462  
Nó 210730 adicionado à tabela, com configuração 731850264  
Nó 210732 adicionado à tabela, com configuração 731850624  
Nó 210734 adicionado à tabela, com configuração 731852460  
Nó 210736 adicionado à tabela, com configuração 731852064  
Nó 210738 adicionado à tabela, com configuração 731852604  
Nó 210740 adicionado à tabela, com configuração 731856420  
Nó 210742 adicionado à tabela, com configuração 731856024  
Nó 210744 adicionado à tabela, com configuração 731856204  
Nó 210746 adicionado à tabela, com configuração 731845062  
Nó 210748 adicionado à tabela, com configuração 731845260  
Nó 210750 adicionado à tabela, com configuração 731845620  
Nó 210752 adicionado à tabela, com configuração 731840562  
Nó 210754 adicionado à tabela, com configuração 731840265  
  
Nó 210725 adicionado à tabela, com configuração 731854602  
Nó 210727 adicionado à tabela, com configuração 731850426  
Nó 210729 adicionado à tabela, com configuração 731850246  
Nó 210731 adicionado à tabela, com configuração 731850642  
Nó 210733 adicionado à tabela, com configuração 731852406  
Nó 210735 adicionado à tabela, com configuração 731852046  
Nó 210737 adicionado à tabela, com configuração 731852640  
Nó 210739 adicionado à tabela, com configuração 731856402  
Nó 210741 adicionado à tabela, com configuração 731856042  
Nó 210743 adicionado à tabela, com configuração 731856240  
Nó 210745 adicionado à tabela, com configuração 731845026  
Nó 210747 adicionado à tabela, com configuração 731845206  
Nó 210749 adicionado à tabela, com configuração 731845602  
Nó 210751 adicionado à tabela, com configuração 731840526  
Nó 210753 adicionado à tabela, com configuração 731840256  
Nó 210755 adicionado à tabela, com configuração 731840652
```

e um módulo para criação de grafos e arestas *uma vez determinados os vizinhos de um nó através de uma função implementada*.

Assim passando por todos os nós (com uma condicional evitando repetição) criamos arestas entre os vizinhos e usamos esta função que determina os vizinhos para mostrar os vizinhos do nó inicial e assim o par (cfg1, cfg2) de nós ligados por uma aresta, e criamos a função *troca_proibida* para simular uma troca entre a peça 4 e 5 no tabuleiro num estado do nó inicial do grafo, o que seria um movimento proibido em um movimento, logo, nós que não são conectados por uma aresta.

```

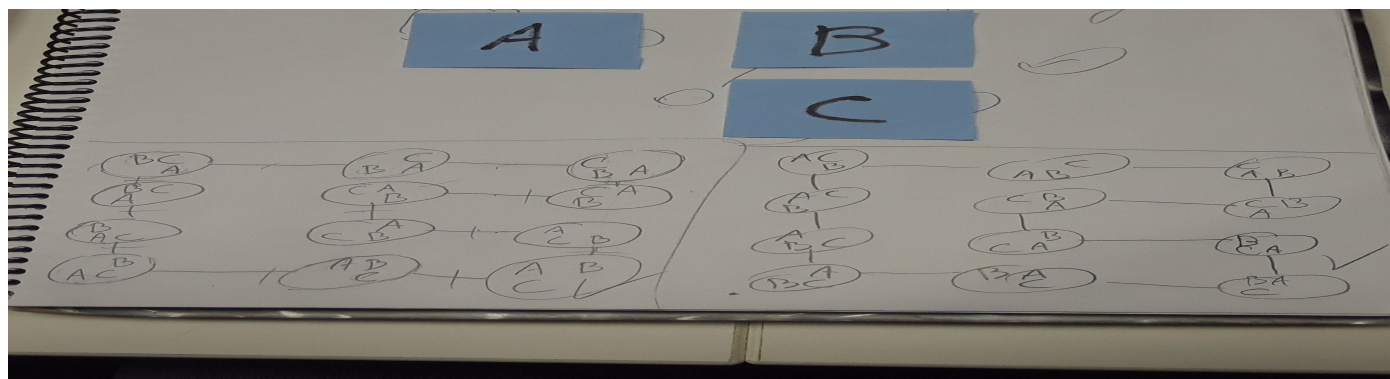
puc@notepuc31: ~
Nó: 362868
Vizinhos: [[0, 5, 4, 2, 6, 3, 8, 1, 7], [2, 5, 4, 8, 6, 3, 0, 1, 7], [2, 5, 4, 6, 0, 3, 8, 1, 7]]
Nó: 362869
Vizinhos: [[0, 5, 4, 2, 6, 1, 7, 3, 8], [2, 5, 4, 7, 6, 1, 0, 3, 8], [2, 5, 4, 6, 0, 1, 7, 3, 8]]
Nó: 362870
Vizinhos: [[0, 5, 4, 2, 6, 1, 7, 8, 3], [2, 5, 4, 7, 6, 1, 0, 8, 3], [2, 5, 4, 6, 0, 1, 7, 8, 3]]
Nó: 362871
Vizinhos: [[0, 5, 4, 2, 6, 1, 3, 7, 8], [2, 5, 4, 3, 6, 1, 0, 7, 8], [2, 5, 4, 6, 0, 1, 3, 7, 8]]
Nó: 362872
Vizinhos: [[0, 5, 4, 2, 6, 1, 3, 8, 7], [2, 5, 4, 3, 6, 1, 0, 8, 7], [2, 5, 4, 6, 0, 1, 3, 8, 7]]
Nó: 362873
Vizinhos: [[0, 5, 4, 2, 6, 1, 8, 7, 3], [2, 5, 4, 8, 6, 1, 0, 7, 3], [2, 5, 4, 6, 0, 1, 8, 7, 3]]
Nó: 362874
Vizinhos: [[0, 5, 4, 2, 6, 1, 8, 3, 7], [2, 5, 4, 8, 6, 1, 0, 3, 7], [2, 5, 4, 6, 0, 1, 8, 3, 7]]
Nó: 362875
Vizinhos: [[0, 5, 4, 2, 6, 8, 7, 3, 1], [2, 5, 4, 7, 6, 8, 0, 3, 1], [2, 5, 4, 6, 0, 8, 7, 3, 1]]
Nó: 362876
Vizinhos: [[0, 5, 4, 2, 6, 8, 7, 1, 3], [2, 5, 4, 7, 6, 8, 0, 1, 3], [2, 5, 4, 6, 0, 8, 7, 1, 3]]
Nó: 362877
Vizinhos: [[0, 5, 4, 2, 6, 8, 3, 7, 1], [2, 5, 4, 3, 6, 8, 0, 7, 1], [2, 5, 4, 6, 0, 8, 3, 7, 1]]
Nó: 362878
Vizinhos: [[0, 5, 4, 2, 6, 8, 3, 1, 7], [2, 5, 4, 3, 6, 8, 0, 1, 7], [2, 5, 4, 6, 0, 8, 3, 1, 7]]
Nó: 362879
Vizinhos: [[0, 5, 4, 2, 6, 8, 1, 7, 3], [2, 5, 4, 1, 6, 8, 0, 7, 3], [2, 5, 4, 6, 0, 8, 1, 7, 3]]
Nó: 362880
Vizinhos: [[0, 5, 4, 2, 6, 8, 1, 3, 7], [2, 5, 4, 1, 6, 8, 0, 3, 7], [2, 5, 4, 6, 0, 8, 1, 3, 7]]
O numero de arestas do grafo é: 1637088

Nó 0 inicial: [7, 3, 1, 8, 6, 0, 4, 5, 2]
Seus nós vizinhos (contém arestas entre si): [[7, 3, 0, 8, 6, 1, 4, 5, 2], [7, 3, 1, 8, 6, 2, 4, 5, 0], [7, 3, 1, 8, 0, 6, 4, 5, 2]]
Nós que não contém arestas entre si: [7, 3, 1, 8, 6, 0, 4, 5, 2] e [7, 3, 1, 8, 6, 0, 5, 4, 2]
(base) puc@notepuc31:~$

```

E assim determinamos que o grafo de 362.880 nós possui no total 1.637.088 arestas.

Dada a demora na (re)criação do grafo em instâncias seguintes o algoritmo para determinação do número de componentes foi adaptado (principalmente a *get_neighbors*) para uma matriz 2x2, como se o jogo tivesse apenas [0,1,2,3] peças com 0 sendo a casa vazia; o qual manualmente pode ser atestado que o grafo possui 2 componentes conexas (a outra obtida através da troca de uma peça por outra, por exemplo '2' por '1', (no papel foi usado 'B' e 'A') um movimento proibido no jogo e assim inacessível a partir de uma componente 'A' 'B', no grafo feito manualmente estes seriam os nós iniciais no canto inferior direito:



Atestamos então 2 componentes de 12 nós cada, exatamente os 4!
Estados possíveis de um jogo de [0,1,2,3] peças.

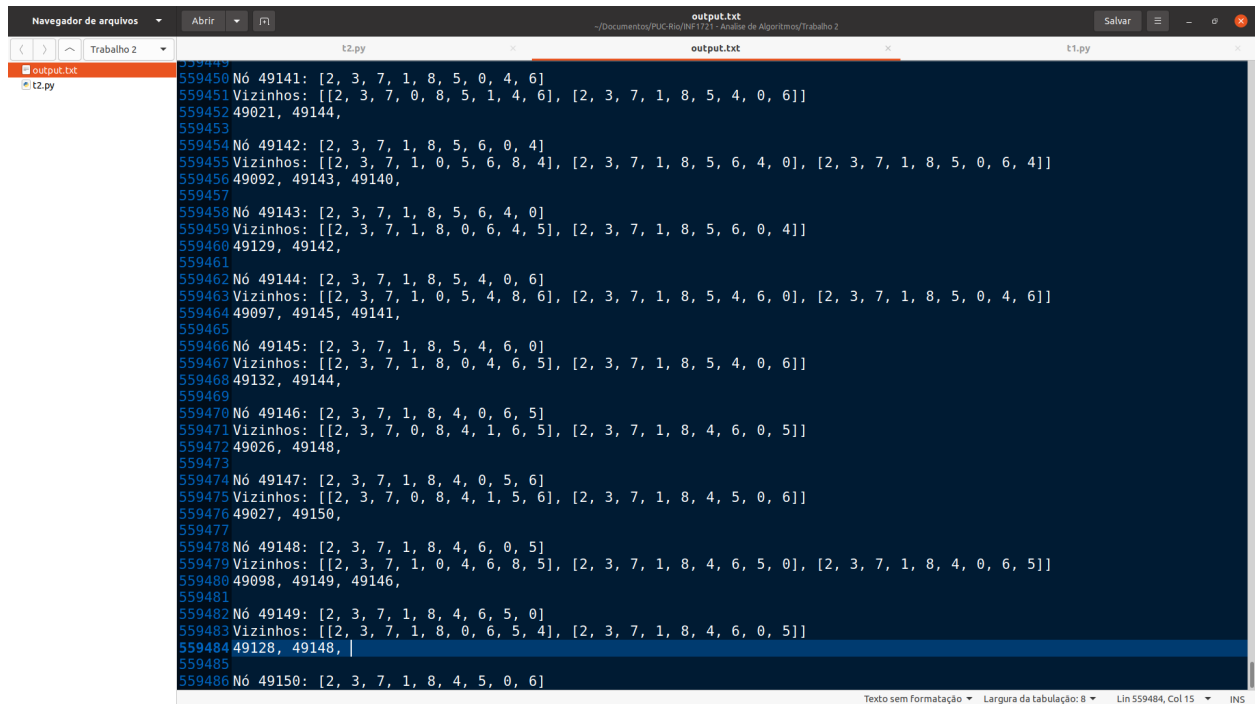
O resultados destes (já que a geração do grafo (agora salvo num arquivo)
de 362.880 nós ainda está há dias sendo gerado...):

```
puc@notepuc31: ~  
ne, None, None]  
  
Nó inicial: 0  
visitados? {0: False, 5: False, 6: False, 1: False, 3: False, 7: False, 2: False, 8: False, 4: False, 10: False, 12: False, 18: Fal  
se, 14: False, 9: False, 15: False, 23: False, 20: False, 11: False, 17: False, 21: False, 13: False, 16: False, 19: False, 22: Fal  
se}  
  
i: 1  
L: [[0], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]  
, None, None]  
i: 2  
L: [[0], [5, 6], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]  
, None, None]  
i: 3  
L: [[0], [5, 6], [4, 12], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]  
, None, None]  
i: 4  
L: [[0], [5, 6], [4, 12], [10, 13], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, N  
one, None, None, None]  
i: 5  
L: [[0], [5, 6], [4, 12], [10, 13], [20, 15], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, Non  
e, None, None, None]  
i: 6  
L: [[0], [5, 6], [4, 12], [10, 13], [20, 15], [22, 9], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]  
, None, None, None]  
i: 7  
L: [[0], [5, 6], [4, 12], [10, 13], [20, 15], [22, 9], [23], [], None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, None]  
, None, None, None]  
  
nós visitados:  
0, 5, 6, 4, 10, 12, 9, 15, 23, 20, 13, 22,  
quantidade de visitados: 12  
(base) puc@notepuc31:~$
```

(as camadas 'None' já foram removidas [verificável] do código, a
pesquisa indicou o uso de uma função queue e dequeue no python mas
procuramos seguir o slide e usar uma variável i na iteração do código, isso
de inicio demandou a criação de um vetor do tamanho do número de nós,
vista que a quantidade de camadas nunca será maior que a de nós. Mas de
novo, isso já foi aperfeiçoado para seguir o modelo original e incremental
da BFS)

É possível observar quais e quantos nós foram visitados em cada
chamada da BFS.

(nesta imagem o print do número de componentes não aparece, mas o
código retorna [verificável] este número corretamente, neste momento o
código gerando o grafo de 362.880 nós está rodando)



```
559449
559450 Nó 49141: [2, 3, 7, 1, 8, 5, 0, 4, 6]
559451 Vizinhos: [[2, 3, 7, 0, 8, 5, 1, 4, 6], [2, 3, 7, 1, 8, 5, 4, 0, 6]]
559452 49021, 49144,
559453
559454 Nó 49142: [2, 3, 7, 1, 8, 5, 6, 0, 4]
559455 Vizinhos: [[2, 3, 7, 1, 0, 5, 6, 8, 4], [2, 3, 7, 1, 8, 5, 6, 4, 0], [2, 3, 7, 1, 8, 5, 0, 6, 4]]
559456 49092, 49143, 49140,
559457
559458 Nó 49143: [2, 3, 7, 1, 8, 5, 6, 4, 0]
559459 Vizinhos: [[2, 3, 7, 1, 8, 0, 6, 4, 5], [2, 3, 7, 1, 8, 5, 6, 0, 4]]
559460 49129, 49142,
559461
559462 Nó 49144: [2, 3, 7, 1, 8, 5, 4, 0, 6]
559463 Vizinhos: [[2, 3, 7, 1, 0, 5, 4, 8, 6], [2, 3, 7, 1, 8, 5, 4, 6, 0], [2, 3, 7, 1, 8, 5, 0, 4, 6]]
559464 49097, 49145, 49141,
559465
559466 Nó 49145: [2, 3, 7, 1, 8, 5, 4, 6, 0]
559467 Vizinhos: [[2, 3, 7, 1, 8, 0, 4, 6, 5], [2, 3, 7, 1, 8, 5, 4, 0, 6]]
559468 49132, 49144,
559469
559470 Nó 49146: [2, 3, 7, 1, 8, 4, 0, 6, 5]
559471 Vizinhos: [[2, 3, 7, 0, 8, 4, 1, 6, 5], [2, 3, 7, 1, 8, 4, 6, 0, 5]]
559472 49026, 49148,
559473
559474 Nó 49147: [2, 3, 7, 1, 8, 4, 0, 5, 6]
559475 Vizinhos: [[2, 3, 7, 0, 8, 4, 1, 5, 6], [2, 3, 7, 1, 8, 4, 5, 0, 6]]
559476 49027, 49150,
559477
559478 Nó 49148: [2, 3, 7, 1, 8, 4, 6, 0, 5]
559479 Vizinhos: [[2, 3, 7, 1, 0, 4, 6, 8, 5], [2, 3, 7, 1, 8, 4, 6, 5, 0], [2, 3, 7, 1, 8, 4, 0, 6, 5]]
559480 49098, 49149, 49146,
559481
559482 Nó 49149: [2, 3, 7, 1, 8, 4, 6, 5, 0]
559483 Vizinhos: [[2, 3, 7, 1, 8, 0, 6, 5, 4], [2, 3, 7, 1, 8, 4, 6, 0, 5]]
559484 49128, 49148, |
559485
559486 Nó 49150: [2, 3, 7, 1, 8, 4, 5, 0, 6]
```

O qual enquanto isso indica o nó, sua configuração e o mesmos dos seus vizinhos.

Por final temos a função `BFS_com_caminho` para encontrar a configuração inicial viável mais difícil de alcançar, lançando uma BFS a partir de `cfg*`; utilizando queue como uma lista de tuplas.