



UNIDADE CURRICULAR: Introdução à Inteligência Artificial

CÓDIGO: 21071

DOCENTE: José Coelho

A preencher pelo estudante

NOME: José Augusto Oliveira Azevedo

N.º DE ESTUDANTE: 2200655

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 15 de Abril de 2024

Critérios	Auto-avaliação:
Análise do Problema (1 valor)	1
Identificação de Algoritmos (1 valor)	1
Resultados (2 valores)	2

Auto-avaliação de acordo com o avaliador: +0.1 na nota do e-fólio

Critérios de correção no enunciado.

TRABALHO / RESOLUÇÃO:

O trabalho pretendido resume-se a dada uma verba que se traduz uma área, encontrar num mapa (matriz) uma área que proteja (contenha) um valor minimo fornecido.

Para a realização deste eFolio decidi efetuar o mesmo usando a linguagem de programação Python.

Dados iniciais:

- Matriz (mapa) com as dimensões N x M
- Verba a gastar com delegacias e deputados
- Objectivo de protecção

Numa análise inicial entendi que era benéfico traduzir o número de Deputados/Raio de protecção em dimensão de matriz e custo associado.

Assim temos a seguinte relação:

Deputados	Raio de protecção	Dimensão da Matriz	Custo
0(apenas o xerife)	1 (protege 3x3)	3 x 3	4
1	2 (protege 5x5)	5 x 5	5
5	3 (protege 7x7)	7 x 7	9
13	5 (protege 9x9)	9 x 9	17

Foi criada a função `combinacoes_delegacias`, aceita como parâmetros `array` `custo_delegacias` que no caso do problema atual é: [4, 5, 9, 17] com os valores de custo; e a verba a utilizar (`valor_alto`); como resultado obtemos todas as combinações possíveis para a verba.

Por exemplo na instância 5, temos uma verba de 12 que resulta nos possíveis gastos:

$$[5], [4], [9], [4, 4] = 8, [5, 5] = 10, [4, 5] = 9, [4, 4, 4] = 12$$

Ver no anexo imagem 1 o resultado do processamento da função.

Na sequencia desta função foi criada a função `troca_custo_por_matriz`. A função é responsável por traduzir os custos anteriormente calculados para as dimensões das áreas

que se pode ter no mapa. Aptei por usar a dimensão total das áreas e não a dimensão do raio, o resultado obtido seria semelhante.

Portanto no exemplo dado da instância 5, com uma verba de 12, os possíveis gastos são:

$$[5], [4], [9], [4, 4] = 8, [5, 5] = 10, [4, 5] = 9, [4, 4, 4] = 12$$

Que resultam nas possíveis dimensões para as áreas das delegacias:

$$[[5], [3], [7], [3, 3], [5, 5], [3, 5], [3, 3, 3]]$$

Ver no anexo imagem 2 o resultado do processamento da função.

Portanto estamos perante um problema de Procura.

Objectivo: Encontrar uma área ou áreas no mapa (depende da verba) que proteja determinado número de pessoas.

Estado inicial: O mapa sem qualquer área seleccionada

Acções: Seleccionar área do mapa com uma dimensão $n \times n$

Mover a área para a direita ou para baixo no mapa até aos seus limites

Solução: Quando encontrar uma área ou áreas no mapa que sejam protejas X pessoas

Perante este problema defini que efetuaria uma representação estruturada do estado. Cada nó (estado) tem a submatriz/área do mapa seleccionada, o total de elementos dessa área, as coordenadas do canto superior esquerdo da área, se é possível ser solução ou não e os nós filhos.

```
class Estado:
    def __init__(self, submatriz, total, top_left, solucao):
        self.submatriz = submatriz
        self.total = total
        self.top_left = top_left
        self.children = []
        self.solucao = solucao
```

Foi implementado um algoritmo de procura cega em profundidade (Depth-first search) pois penso que dá melhor resposta ao pretendido. Ver função no Anexo.

Pseudo-código da função:

```
função depth_first_search(matrix, dimensions, nivel, alvo):
    encontrou = Falso

    se o comprimento de dimensions ≠ 0 então:
```

```

n m := comprimento da matriz
original_matrix_copy := fazer uma cópia da matriz atual
working_matrix := fazer uma cópia da matriz atual para trabalhar

para cada dimensão i em dimensions[nivel] faça:
    para cada linha r na matriz, com índice de 0 até n - i + 1 faça:
        para cada coluna c na matriz, com índice de 0 até m - i + 1
faça:
    submatriz := uma nova matriz com os elementos da matriz[r:r
+ i][c:c + i]
    verificar se há algum elemento igual a -1 na submatriz
    se não houver -1 na submatriz então:
        total := a soma de todos os elementos da submatriz
        top_left := a tupla (r, c)
        node := um novo estado com a submatriz, total, top_left
e False como valores
        n_geracoes := n_geracoes + 1

        se total ≥ alvo então:
            encontrou := Verdadeiro
            node.solucao := Verdadeiro

        para cada elemento (x, y) na submatriz faça:
            working_matrix[x][y] := -1 ou 0
            -1 não há sobreposicao 0 há sobreposicao

        se ((o comprimento de dimensions - 1) ≠ nivel) e
(encontrou é Falso) então:
            nivel := nivel + 1
            n_expansoes := n_expansoes + 1

            node.children, a_expansoes, a_geracoes, encontrou :=
depth_first_search(working_matrix, dimensions, nivel, alvo - total)
            n_geracoes := n_geracoes + a_geracoes
            n_expansoes := n_expansoes + a_expansoes
            node.solucao := encontrou
            nivel := nivel - 1

        se node.solucao então:
            adicionar node à lista root_nodes

        se encontrou então:
            retornar root_nodes, n_expansoes, n_geracoes,
Verdadeiro

    working_matrix := fazer uma cópia da matriz original

retornar root_nodes, n_expansoes, n_geracoes, encontrou

```

Considerarei como expansão quando se cria uma nova posição no mapa e geração ao conjunto das diferentes combinações de posicoes possiveis dentro da matriz/mapa – todas as combinações testadas.

Estas decisões levaram a que a profundidade da arvore de procura fosse no máximo de 6 niveis. Podemos ver no anexo imagem 3 a ilustração da estrutura da árvore e da procura.

Encontrei os seguintes resultados:

ID	N	M	Verba	Proteger A	Proteger B
1	5	5	4	19	20
2	5	5	4	21	22
3	7	7	8	67	68
4	7	7	8	59	60
5	9	9	12	125	126
6	9	9	12	57	58
7	11	11	16	140	141
8	11	11	16	93	94
9	13	13	20	211	212
10	13	13	20	125	126

A verde onde foram encontradas soluções e a laranja não foi encontrada solução.

Quadro de resultados obtidos

Instância	1a	1b	2a	2b
Expansões	0	0	0	0
Gerações	7	9	3	9
Resultado	Encontrou	Não	Encontrou	Não
Temp (msec)	0,00	0,00	0,00	1,00
Instância	3a	3b	4a	4b
Expansões	0	25	0	25
Gerações	27	684	33	684
Resultado	Encontrou	Não	Encontrou	Não
Temp (msec)	1,00	6,19	1,00	8,14
Instância	5a	5b	6a	6b
Expansões	0	2573	55	2573
Gerações	75	124556	2689	124556
Resultado	Encontrou	Não	Encontrou	Não
Temp (msec)	1,00	763,2	24,08	774,45
Instância	7a	7b	8a	8b
Expansões	836	558208	356	363
Gerações	40937	44786325	17365	17715
Resultado	Encontrou	Não	Encontrou	Encontrou
Temp (msec)	365,19	296597,57	139,48	165,61
Instância	9a	9b	10a	10b
Expansões	31157	31490	22426	25780
Gerações	2612021	2628475	2180669	2346396
Resultado	Encontrou	Encontrou	Encontrou	Encontrou
Temp (msec)	22138,32	24014,81	16508,65	19782,91

ANEXOS

Instância	Verba	Custos possíveis
1	4	[[4]]
2	4	[[4]]
3	8	[[4], [5], [4, 4]]
4	8	[[4], [5], [4, 4]]
5	12	[[5], [4], [9], [4, 4], [5, 5], [4, 5], [4, 4, 4]]
6	12	[[5], [4], [9], [4, 4], [5, 5], [4, 5], [4, 4, 4]]
7	16	[[5], [4], [9], [4, 4], [5, 5], [4, 9], [4, 5], [5, 9], [5, 5, 5], [4, 4, 5], [4, 4, 4], [4, 5, 5], [4, 4, 4, 4]]
8	16	[[5], [4], [9], [4, 4], [5, 5], [4, 9], [4, 5], [5, 9], [5, 5, 5], [4, 4, 5], [4, 4, 4], [4, 5, 5], [4, 4, 4, 4]]
9	20	[[5], [17], [4], [9], [4, 9], [4, 5], [5, 9], [4, 4], [5, 5], [9, 9], [4, 4, 4], [4, 5, 9], [5, 5, 9], [4, 5, 5], [4, 4, 9], [5, 5, 5], [4, 4, 5], [4, 4, 4, 4], [4, 4, 5, 5], [4, 5, 5, 5], [5, 5, 5, 5], [4, 4, 4, 5], [4, 4, 4, 4, 4]]
10	20	[[5], [17], [4], [9], [4, 9], [4, 5], [5, 9], [4, 4], [5, 5], [9, 9], [4, 4, 4], [4, 5, 9], [5, 5, 9], [4, 5, 5], [4, 4, 9], [5, 5, 5], [4, 4, 5], [4, 4, 4, 4], [4, 4, 5, 5], [4, 5, 5, 5], [5, 5, 5, 5], [4, 4, 4, 5], [4, 4, 4, 4, 4]]

Imagem 1: Resultado do processamento da função combinacoes_delegacias

Instância	Verba	Areas possíveis
1	3	[[3]]
2	3	[[3]]
3	8	[[3], [5], [3, 3]]
3	8	[[3], [5], [3, 3]]
5	12	[[5], [3], [7], [3, 3], [5, 5], [3, 5], [3, 3, 3]]
6	12	[[5], [3], [7], [3, 3], [5, 5], [3, 5], [3, 3, 3]]
7	16	[[5], [3], [7], [3, 3], [5, 5], [3, 7], [3, 5], [5, 7], [5, 5, 5], [3, 3, 5], [3, 3, 3], [3, 5, 5], [3, 3, 3, 3]]
8	16	[[5], [3], [7], [3, 3], [5, 5], [3, 7], [3, 5], [5, 7], [5, 5, 5], [3, 3, 5], [3, 3, 3], [3, 5, 5], [3, 3, 3, 3]]
7	20	[[5], [9], [3], [7], [3, 7], [3, 5], [5, 7], [3, 3], [5, 5], [7, 7], [3, 3, 3], [3, 5, 7], [5, 5, 7], [3, 5, 5], [3, 3, 7], [5, 5, 5], [3, 3, 5], [3, 3, 3, 3], [3, 3, 5, 5], [3, 5, 5, 5], [5, 5, 5, 5], [3, 3, 3, 5], [3, 3, 3, 3, 3]]
10	20	[[5], [9], [3], [7], [3, 7], [3, 5], [5, 7], [3, 3], [5, 5], [7, 7], [3, 3, 3], [3, 5, 7], [5, 5, 7], [3, 5, 5], [3, 3, 7], [5, 5, 5], [3, 3, 5], [3, 3, 3, 3], [3, 3, 5, 5], [3, 5, 5, 5], [5, 5, 5, 5], [3, 3, 3, 5], [3, 3, 3, 3, 3]]

Imagem 2: Resultado do processamento da função troca_custo_por_matriz

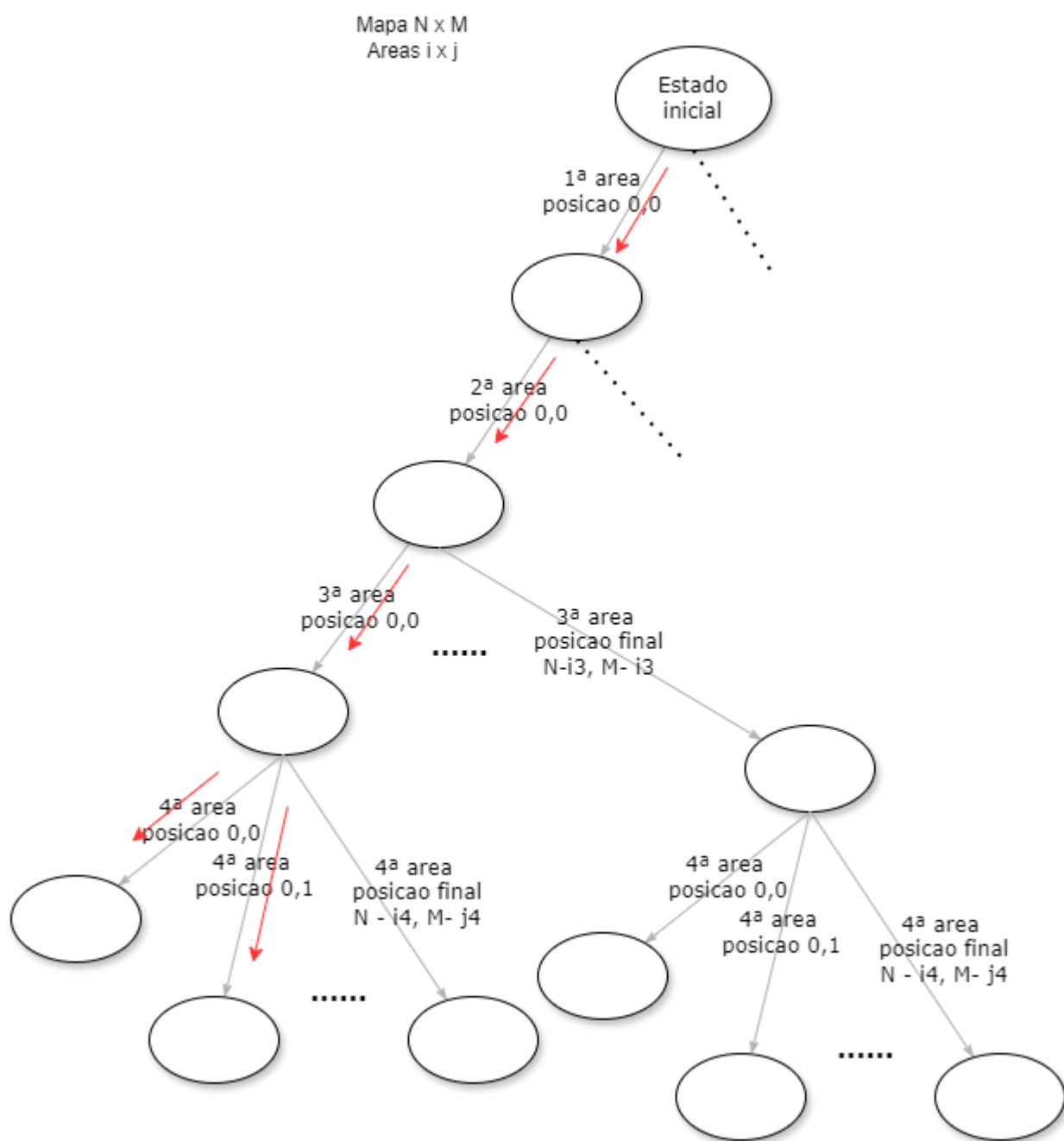


Imagem 3: Ilustração da estrutura da árvore e da procura

Resultados:

Instância 1 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou	
1	19	[3]	0	7	0.00	1	
Fim: 1	19		0	7	0.00	['(2, 0) 3']	

Populacao:

[1, 0, 0]

[4, 4, 1]

[6, 0, 3]

Total: 19, Top-left: (2, 0) , Center: 4, 2, Raio: 1 , Solucao: True

0	7	0	0	4	
0	0	0	4	0	
1	0	0	0	0	
4	4	1	0	0	
6	0	3	4	4	

Disponivel: 0 [1 delegacias colocadas]

Protegidos: 19 / 19

Solucao!

Instância 1 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou	
1	20	[3]	0	9	0.00	0	
Fim: 1	20		0	9	0.00		
Sem Solução !							

ID 1

Proteger A - 19

	7			4
			4	
1				
4	4	1		
6		3	4	4

Proteger B - 20 - sem solução

	7			4
			4	
1				
4	4	1		
6		3	4	4

Instância 2 a

```
-----
| Instância | Alvo      | Dimensões | Expansões | Gerações | Tempo Process.(ms) | Encontrou |
-----
| 2         | 21        | [3]       | 0         | 3        | 0.00            | 1         |
-----
| Fim: 2    | 21        |           | 0         | 3        | 0.00            | ['(0, 2) 3'] |
-----

Populacao:
[0, 10, 1]
[0, 0, 0]
[1, 6, 3]
Total: 21, Top-left: (0, 2) , Center: 2, 4, Raio: 1 , Solucao: True

-----
| 4  0  0  10  1 |
| 1  0  0  0  0 |
| 0  0  1  6  3 |
| 0  4  0  0  2 |
| 8  0  6  3  0 |
-----
Disponivel: 0 [ 1 delegacias colocadas]
Protegidos: 21 / 21
Solucao!
```

Instância 2 b

```
-----
| Instância | Alvo      | Dimensões | Expansões | Gerações | Tempo Process.(ms) | Encontrou |
-----
| 2         | 22        | [3]       | 0         | 9        | 1.00            | 0         |
-----
| Fim: 2    | 22        |           | 0         | 9        | 1.00            |           |
-----

Sem Solução !
```

ID 2

Proteger A - 21

4			10	1
1				
		1	6	3
	4			
8		6	3	

Proteger B - 22 - sem solução

4			10	1
1				
		1	6	3
	4			
8		6	3	

Instância 3 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
3	67	[3]	0	25	1.00	0
3	67	[5]	0	2	0.00	1
Fim: 3	67		0	27	1.00	['(0, 1) 5']

Populacao:
 [8, 0, 4, 5, 10]
 [4, 0, 7, 0, 4]
 [2, 4, 2, 0, 0]
 [7, 0, 1, 2, 0]
 [4, 0, 0, 3, 0]
 Total: 67, Top-left: (0, 1) , Center: 3, 4, Raio: 2 , Solucao: True

```

-----
| 0 8 0 4 5 10 0 |
| 0 4 0 7 0 4 0 |
| 0 2 4 2 0 0 2 |
| 0 7 0 1 2 0 0 |
| 2 4 0 0 3 0 2 |
| 0 4 0 0 3 0 0 |
| 2 0 0 0 0 0 0 |
-----
Disponivel: 3 [ 1 delegacias colocadas]
Protegidos: 67 / 67
Solucao!

```

Instância 3 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
3	68	[3]	0	25	1.00	0
3	68	[5]	0	9	0.00	0
3	68	[3, 3]	25	650	5.19	0
Fim: 3	68		25	684	6.19	

Sem Solução !

ID 3 **Proteger A - 67** **67**

	8		4	5	10	
	4		7		4	
	2	4	2			2
	7		1	2		
2	4			3		2
	4			3		
2						

Proteger B - 68 - sem solução

	8		4	5	10	
	4		7		4	
	2	4	2			2
	7		1	2		
2	4			3		2
	4			3		
2						

Instância 4 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
4	59	[3]	0	25	1.00	0
4	59	[5]	0	8	0.00	1
Fim: 4	59		0	33	1.00	['(2, 1) 5']
Populacao:						
[0, 0, 0, 2, 0]						
[1, 0, 8, 5, 7]						
[4, 0, 3, 0, 0]						
[0, 0, 3, 2, 4]						
[8, 3, 6, 3, 0]						
Total: 59, Top-left: (2, 1) , Center: 5, 4, Raio: 2 , Solucao: True						

0	0	1	0	7	0	1
0	1	4	0	0	0	4
0	0	0	0	2	0	0
3	1	0	8	5	7	7
0	4	0	3	0	0	0
0	0	0	3	2	4	2
0	8	3	6	3	0	0

Disponivel: 3 [1 delegacias colocadas]						
Protegidos: 59 / 59						
Solucao!						

Instância 4 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
4	60	[3]	0	25	0.00	0
4	60	[5]	0	9	0.00	0
4	60	[3, 3]	25	650	8.14	0
Fim: 4	60		25	684	8.14	
Sem Solução !						

ID 4 Proteger A - 59 59

		1		7		1
	1	4				4
				2		
3	1		8	5	7	7
	4		3			
			3	2	4	2
	8	3	6	3		

Proteger B - 60 - sem solução

		1		7		1
	1	4				4
				2		
3	1		8	5	7	7
	4		3			
			3	2	4	2
	8	3	6	3		

Instância 5 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
5	125	[5]	0	25	0.00	0
5	125	[3]	0	49	1.00	0
5	125	[7]	0	1	0.00	1
Fim: 5	125		0	75	1.00	['(0, 0) 7']
Populacao:						
[6, 7, 2, 0, 0, 0, 0]						
[3, 3, 6, 0, 8, 4, 3]						
[0, 0, 8, 0, 0, 0, 2]						
[0, 0, 0, 1, 0, 3, 2]						
[0, 0, 0, 7, 4, 0, 1]						
[12, 8, 0, 5, 4, 1, 4]						
[8, 0, 1, 2, 4, 3, 3]						
Total: 125, Top-left: (0, 0) , Center: 4, 4, Raio: 3 , Solucao: True						

6	7	2	0	0	0	0
3	3	6	0	8	4	3
0	0	8	0	0	0	2
0	0	0	1	0	3	2
0	0	0	7	4	0	1
12	8	0	5	4	1	4
8	0	1	2	4	3	3
1	1	0	0	0	5	0
4	0	0	0	4	6	13

Disponivel: 3 [1 delegacias colocadas]						
Protegidos: 125 / 125						
Solucao!						

Instância 5 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
5	126	[5]	0	25	1.01	0
5	126	[3]	0	49	0.00	0
5	126	[7]	0	9	0.00	0
5	126	[3, 3]	49	2450	18.09	0
5	126	[5, 5]	25	650	6.00	0
5	126	[3, 5]	49	1274	11.01	0
5	126	[3, 3, 3]	2450	120099	727.08	0
Fim: 5	126		2573	124556	763.20	
Sem Solução !						

ID 5	Proteger A - 125							125
6	7	2						
3	3	6		8	4	3	1	
		8				2	4	
			1		3	2		
			7	4		1		
12	8		5	4	1	4	3	4
8		1	2	4	3	3		
1	1					5		
4				4	6		13	2

Proteger B - 126 - sem solução								
6	7	2						
3	3	6		8	4	3	1	
		8				2	4	
			1		3	2		
			7	4		1		
12	8		5	4	1	4	3	4
8		1	2	4	3	3		
1	1					5		
4				4	6		13	2

Instância 6 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
6	57	[5]	0	25	1.00	0
6	57	[3]	0	49	0.00	0
6	57	[7]	0	9	1.06	0
6	57	[3, 3]	49	2450	21.02	0
6	57	[5, 5]	6	156	1.00	1
Fim: 6	57		55	2689	24.08	[['(4, 4) 5', '(1, 0) 5']]
Populacao:						
[4, 0, 8, 4, 0]						
[0, 0, 0, 0, 0]						
[0, 0, 0, 0, 3]						
[0, 3, 0, 0, 0]						
[0, 0, 0, 1, 1]						
Total: 24, Top-left: (1, 0) , Center: 4, 3, Raio: 2 , Solucao: True						
Populacao:						
[0, 0, 0, 0, 0]						
[0, 0, 0, 3, 0]						
[0, 0, 0, 1, 0]						
[8, 0, 4, 3, 10]						
[0, 4, 0, 0, 0]						
Total: 33, Top-left: (4, 4) , Center: 7, 7, Raio: 2 , Solucao: True						
Disponivel: 2 [2 delegacias colocadas]						
Protegidos: 57 / 57						
Solucao!						

Instância 6 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
6	58	[5]	0	25	0.00	0
6	58	[3]	0	49	1.00	0
6	58	[7]	0	9	0.00	0
6	58	[3, 3]	49	2450	17.13	0
6	58	[5, 5]	25	650	6.01	0
6	58	[3, 5]	49	1274	9.00	0
6	58	[3, 3, 3]	2450	120099	741.31	0
Fim: 6	58		2573	124556	774.45	
Sem Solução !						

ID 6

Proteger A - 57

57

4		8	4					
				3			1	
	3							
			1	1			3	
		2	4				1	
	2			8		4	3	10
		3			4			

Proteger B - 58 - sem solução

4		8	4					
				3			1	
	3							
			1	1			3	
		2	4				1	
	2			8		4	3	10
		3			4			

Instância 7 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
7	140	[5]	0	49	0.00	0
7	140	[3]	0	81	1.00	0
7	140	[7]	0	25	1.15	0
7	140	[3, 3]	81	6642	43.03	0
7	140	[5, 5]	49	2450	19.01	0
7	140	[3, 7]	81	2106	22.02	0
7	140	[3, 5]	81	4050	32.04	0
7	140	[5, 7]	49	1274	13.01	0
7	140	[5, 5, 5]	495	24260	233.93	1
Fim: 7	140		836	40937	365.19	[[['(6, 6) 5'], '(6, 1) 5'], '(1, 2) 5']]

Populacao:

[11, 2, 0, 0, 9]

[0, 3, 1, 0, 2]

[2, 3, 0, 4, 0]

[0, 0, 4, 0, 1]

[0, 7, 4, 0, 1]

Total: 54, Top-left: (1, 2) , Center: 4, 5, Raio: 2 , Solucao: True

Populacao:

[8, 0, 0, 0, 0]

[3, 0, 0, 5, 2]

[0, 0, 3, 1, 0]

[3, 4, 0, 7, 0]

[2, 0, 4, 0, 3]

Total: 45, Top-left: (6, 1) , Center: 9, 4, Raio: 2 , Solucao: True

Populacao:

[3, 0, 1, 0, 3]

[3, 0, 0, 0, 2]

[2, 8, 0, 0, 0]

[0, 7, 0, 0, 0]

[0, 0, 5, 7, 0]

Total: 41, Top-left: (6, 6) , Center: 9, 9, Raio: 2 , Solucao: True

0	0	0	0	0	3	0	0	0	0	0
0	0	11	2	0	0	9	3	0	0	3
0	0	0	3	1	0	2	0	0	0	0
4	1	2	3	0	4	0	0	4	0	0
5	0	0	0	4	0	1	0	4	3	0
0	0	0	7	4	0	1	0	0	7	0
0	8	0	0	0	0	3	0	1	0	3
0	3	0	0	5	2	3	0	0	0	2
0	0	0	3	1	0	2	8	0	0	0
0	3	4	0	7	0	0	7	0	0	0
4	2	0	4	0	3	0	0	5	7	0

Disponivel: 1 [3 delegacias colocadas]

Protegidos: 140 / 140

Solucao!

Instância 7 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
7	141	[5]	0	49	0.00	0
7	141	[3]	0	81	1.04	0
7	141	[7]	0	25	0.00	0
7	141	[3, 3]	81	6642	45.03	0
7	141	[5, 5]	49	2450	21.01	0
7	141	[3, 7]	81	2106	22.02	0
7	141	[3, 5]	81	4050	32.20	0
7	141	[5, 7]	49	1274	14.06	0
7	141	[5, 5, 5]	2450	120099	1081.92	0
7	141	[3, 3, 5]	6642	328131	2904.60	0
7	141	[3, 3, 3]	6642	538083	3464.84	0
7	141	[3, 5, 5]	4050	198531	1917.68	0
7	141	[3, 3, 3, 3]	538083	43584804	287092.16	0
Fim: 7	141		558208	44786325	296597.57	
Sem Solução !						

ID 7

Proteger A - 140

140

					3					
		11	2		9	3				3
			3	1	2					
4	1	2	3	4			4			
5				4	1		4	3		
			7	4	1			7		
	8				3		1		3	
	3			5	2	3				2
			3	1	2	8				
	3	4		7		7				
4	2		4		3		5	7		

Proteger B - 141 - sem solução

					3					
		11	2		9	3				3
			3	1	2					
4	1	2	3		4			4		
5				4	1			4	3	
			7	4	1			7		
	8				3		1		3	
	3			5	2	3				2
			3	1	2	8				
	3	4		7		7				
4	2		4		3			5	7	

Instância 8 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
8	93	[5]	0	49	1.00	0
8	93	[3]	0	81	1.00	0
8	93	[7]	0	25	0.00	0
8	93	[3, 3]	81	6642	42.35	0
8	93	[5, 5]	49	2450	19.07	0
8	93	[3, 7]	81	2106	22.03	0
8	93	[3, 5]	81	4050	32.02	0
8	93	[5, 7]	49	1274	12.99	0
8	93	[5, 5, 5]	15	688	7.00	1
Fim: 8	93		356	17365	139.48	[[['(5, 0) 5'], '(1, 6) 5'], '(0, 0) 5']
Populacao:						
[1, 0, 0, 0, 0]						
[0, 0, 0, 0, 0]						
[0, 0, 10, 10, 0]						
[0, 4, 1, 0, 8]						
[8, 0, 0, 0, 0]						
Total: 42, Top-left: (0, 0) , Center: 3, 3, Raio: 2 , Solucao: True						
Populacao:						
[0, 0, 0, 0, 0]						
[0, 4, 5, 0, 0]						
[0, 0, 0, 0, 5]						
[6, 0, 0, 0, 0]						
[0, 0, 2, 0, 3]						
Total: 25, Top-left: (1, 6) , Center: 4, 9, Raio: 2 , Solucao: True						
Populacao:						
[0, 0, 0, 0, 13]						
[0, 0, 0, 0, 4]						
[0, 0, 0, 0, 0]						
[0, 0, 4, 0, 0]						
[4, 1, 0, 0, 0]						
Total: 26, Top-left: (5, 0) , Center: 8, 3, Raio: 2 , Solucao: True						

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	10	10	0	0	0	4	5	0	0	0
0	4	1	0	8	0	0	0	0	0	5	0
8	0	0	0	0	0	0	6	0	0	0	0
0	0	0	0	13	0	0	0	2	0	3	0
0	0	0	0	4	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	3	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Disponivel: 1 [3 delegacias colocadas]											
Protegidos: 93 / 93											
Solucao!											

Instância 8 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
8	94	[5]	0	49	1.00	0
8	94	[3]	0	81	0.00	0
8	94	[7]	0	25	0.00	0
8	94	[3, 3]	81	6642	40.49	0
8	94	[5, 5]	49	2450	20.06	0
8	94	[3, 7]	81	2106	27.01	0
8	94	[3, 5]	81	4050	44.02	0
8	94	[5, 7]	49	1274	21.02	0
8	94	[5, 5, 5]	22	1038	11.01	1
Fim: 8	94		363	17715	165.61	[[['(5, 0) 5'], '(2, 6) 5'], '(0, 0) 5']]

Populacao:
[1, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 10, 10, 0]
[0, 4, 1, 0, 8]
[8, 0, 0, 0, 0]
Total: 42, Top-left: (0, 0) , Center: 3, 3, Raio: 2 , Solucao: True
Populacao:
[0, 4, 5, 0, 0]
[0, 0, 0, 0, 5]
[6, 0, 0, 0, 0]
[0, 0, 2, 0, 3]
[0, 0, 0, 1, 0]
Total: 26, Top-left: (2, 6) , Center: 5, 9, Raio: 2 , Solucao: True
Populacao:
[0, 0, 0, 0, 13]
[0, 0, 0, 0, 4]
[0, 0, 0, 0, 0]
[0, 0, 4, 0, 0]
[4, 1, 0, 0, 0]
Total: 26, Top-left: (5, 0) , Center: 8, 3, Raio: 2 , Solucao: True

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	10	10	0	0	0	4	5	0	0	0
0	4	1	0	8	0	0	0	0	0	5	0
8	0	0	0	0	0	6	0	0	0	0	0
0	0	0	0	13	0	0	0	2	0	3	0
0	0	0	0	4	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	3	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Disponivel: 1 [3 delegacias colocadas]
Protegidos: 94 / 94
Solucao!

ID 8

Proteger A - 93					93				
1									
		10	10				4	5	
	4	1		8					5
8					6				
				13			2		3
				4					1
		4					3		
4	1								

Proteger B - 94 - sem solução

94									
1									
		10	10				4	5	
	4	1		8					5
8							6		
				13				2	3
				4					1
		4					3		
4	1								

Instância 9 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
9	211	[5]	0	81	0.99	0
9	211	[9]	0	25	0.00	0
9	211	[3]	0	121	2.15	0
9	211	[7]	0	49	1.02	0
9	211	[3, 7]	121	6050	75.03	0
9	211	[3, 5]	121	9922	79.56	0
9	211	[5, 7]	81	4050	48.08	0
9	211	[3, 3]	121	14762	124.09	0
9	211	[5, 5]	81	6642	54.04	0
9	211	[7, 7]	49	2450	27.02	0
9	211	[3, 3, 3]	14762	1786323	12741.44	0
9	211	[3, 5, 7]	9922	490171	5557.72	0
9	211	[5, 5, 7]	5899	291375	3426.19	1
Fim: 9	211		31157	2612021	22138.32	[[['(0, 1) 7'], '(8, 3) 5'], '(7, 8) 5']

```

Populacao:
[2, 0, 1, 0, 3]
[0, 2, 2, 0, 8]
[1, 0, 0, 0, 7]
[3, 4, 0, 9, 0]
[0, 0, 0, 3, 11]
Total: 56, Top-left: (7, 8) , Center: 10, 11, Raio: 2 , Solucao: True
Populacao:
[0, 0, 6, 0, 0]
[2, 0, 0, 6, 4]
[11, 0, 0, 0, 0]
[0, 1, 4, 3, 4]
[7, 7, 0, 0, 2]
Total: 57, Top-left: (8, 3) , Center: 11, 6, Raio: 2 , Solucao: True
Populacao:
[4, 0, 0, 6, 7, 3, 4]
[0, 2, 0, 3, 0, 0, 6]
[3, 0, 8, 0, 0, 2, 0]
[0, 0, 0, 0, 0, 0, 0]
[6, 0, 8, 0, 3, 0, 0]
[3, 0, 2, 0, 0, 9, 0]
[9, 4, 0, 0, 2, 4, 0]
Total: 98, Top-left: (0, 1) , Center: 4, 5, Raio: 3 , Solucao: True

```

```

-----
| 2  4  0  0  6  7  3  4  0  0  3  0  1  |
| 0  0  2  0  3  0  0  6  0  0  8  11  3  |
| 0  3  0  8  0  0  2  0  0  0  0  0  4  |
| 2  0  0  0  0  0  0  0  0  3  2  0  0  |
| 0  6  0  8  0  3  0  0  0  0  0  0  1  |
| 0  3  0  2  0  0  9  0  0  0  0  5  6  |
| 1  9  4  0  0  2  4  0  0  0  3  2  0  |
| 2  3  0  4  0  0  0  6  2  0  1  0  3  |
| 0  0  0  0  0  6  0  0  0  2  2  0  8  |
| 7  2  4  2  0  0  6  4  1  0  0  0  7  |
| 0  0  0  11  0  0  0  0  3  4  0  9  0  |
| 0  0  0  0  1  4  3  4  0  0  0  3  11  |
| 0  0  4  7  7  0  0  2  0  2  5  0  1  |
-----
Disponivel: 1 [ 3 delegacias colocadas]
Protegidos: 211 / 211
Solucao!

```

Instância 9 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
9	212	[5]	0	81	1.05	0
9	212	[9]	0	25	0.00	0
9	212	[3]	0	121	1.00	0
9	212	[7]	0	49	0.00	0
9	212	[3, 7]	121	6050	65.04	0
9	212	[3, 5]	121	9922	81.06	0
9	212	[5, 7]	81	4050	43.40	0
9	212	[3, 3]	121	14762	147.80	0
9	212	[5, 5]	81	6642	58.08	0
9	212	[7, 7]	49	2450	26.02	0
9	212	[3, 3, 3]	14762	1786323	12940.66	0
9	212	[3, 5, 7]	9922	490171	6764.97	0
9	212	[5, 5, 7]	6232	307829	3884.73	1
Fim: 9	212		31490	2628475	24014.81	[[['(0, 1) 7'], '(8, 8) 5'], '(8, 3) 5']

```
Populacao:
[0, 0, 6, 0, 0]
[2, 0, 0, 6, 4]
[11, 0, 0, 0, 0]
[0, 1, 4, 3, 4]
[7, 7, 0, 0, 2]
Total: 57, Top-left: (8, 3) , Center: 11, 6, Raio: 2 , Solucao: True
Populacao:
[0, 2, 2, 0, 8]
[1, 0, 0, 0, 7]
[3, 4, 0, 9, 0]
[0, 0, 0, 3, 11]
[0, 2, 5, 0, 1]
Total: 58, Top-left: (8, 8) , Center: 11, 11, Raio: 2 , Solucao: True
Populacao:
[4, 0, 0, 6, 7, 3, 4]
[0, 2, 0, 3, 0, 0, 6]
[3, 0, 8, 0, 0, 2, 0]
[0, 0, 0, 0, 0, 0, 0]
[6, 0, 8, 0, 3, 0, 0]
[3, 0, 2, 0, 0, 9, 0]
[9, 4, 0, 0, 2, 4, 0]
Total: 98, Top-left: (0, 1) , Center: 4, 5, Raio: 3 , Solucao: True
```

```
-----
| 2  4  0  0  6  7  3  4  0  0  3  0  1  |
| 0  0  2  0  3  0  0  6  0  0  8  11  3  |
| 0  3  0  8  0  0  2  0  0  0  0  0  4  |
| 2  0  0  0  0  0  0  0  0  3  2  0  0  |
| 0  6  0  8  0  3  0  0  0  0  0  0  1  |
| 0  3  0  2  0  0  9  0  0  0  0  5  6  |
| 1  9  4  0  0  2  4  0  0  0  3  2  0  |
| 2  3  0  4  0  0  0  6  2  0  1  0  3  |
| 0  0  0  0  0  6  0  0  0  2  2  0  8  |
| 7  2  4  2  0  0  6  4  1  0  0  0  7  |
| 0  0  0  11  0  0  0  0  3  4  0  9  0  |
| 0  0  0  0  1  4  3  4  0  0  0  3  11  |
| 0  0  4  7  7  0  0  2  0  2  5  0  1  |
-----
Disponivel: 1 [ 3 delegacias colocadas]
Protegidos: 213 / 212
Solucao!
```

ID 9	Proteger A - 211								211	98	56	57	
	2	4			6	7	3	4			3		1
			2		3			6			8	11	3
		3		8			2						4
	2									3	2		
		6		8		3							1
		3		2			9					5	6
	1	9	4			2	4				3	2	
	2	3		4				6	2		1		3
						6				2	2		8
	7	2	4	2			6	4	1				7
				11					3	4		9	
					1	4	3	4				3	11
			4	7	7			2		2	5		1

Proteger B - 212 - sem solução									213	98	58	57
2	4			6	7	3	4			3		1
		2		3			6			8	11	3
	3		8			2						4
2									3	2		
	6		8		3							1
	3	2				9					5	6
1	9	4			2	4				3	2	
2	3		4				6	2		1		3
					6				2	2		8
7	2	4	2			6	4	1				7
			11					3	4		9	
				1	4	3	4				3	11
		4	7	7			2		2	5		1

Instância 10 a

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
10	125	[5]	0	81	2.00	0
10	125	[9]	0	25	0.00	0
10	125	[3]	0	121	1.00	0
10	125	[7]	0	49	1.13	0
10	125	[3, 7]	121	6050	73.67	0
10	125	[3, 5]	121	9922	82.10	0
10	125	[5, 7]	81	4050	43.05	0
10	125	[3, 3]	121	14762	114.79	0
10	125	[5, 5]	81	6642	68.12	0
10	125	[7, 7]	49	2450	29.02	0
10	125	[3, 3, 3]	14762	1786323	11997.09	0
10	125	[3, 5, 7]	7090	350194	4096.66	1
Fim: 10	125		22426	2180669	16508.65	[[['(0, 5) 7'], '(4, 0) 5'], '(7, 9) 3']]

```

Populacao:
[9, 0, 0]
[3, 4, 0]
[5, 0, 1]
Total: 22, Top-left: (7, 9) , Center: 9, 11, Raio: 1 , Solucao: True
Populacao:
[0, 6, 9, 0, 0]
[0, 0, 0, 0, 0]
[0, 3, 0, 0, 0]
[0, 0, 0, 1, 3]
[9, 0, 0, 3, 3]
Total: 37, Top-left: (4, 0) , Center: 7, 3, Raio: 2 , Solucao: True
Populacao:
[0, 9, 0, 0, 0, 12, 0]
[0, 0, 0, 0, 1, 0, 0]
[0, 2, 0, 0, 2, 0, 0]
[9, 4, 0, 0, 0, 6, 0]
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 6, 10, 0, 1]
[1, 0, 0, 0, 0, 0, 2]
Total: 66, Top-left: (0, 5) , Center: 4, 9, Raio: 3 , Solucao: True

```

```
-----
| 0 0 1 4 0 0 9 0 0 0 12 0 1 |
| 0 0 0 0 0 0 0 0 0 0 1 0 0 0 |
| 1 0 0 0 0 0 2 0 0 2 0 0 0 |
| 0 0 0 0 0 9 4 0 0 0 6 0 0 |
| 0 6 9 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 1 6 10 0 1 4 |
| 0 3 0 0 0 1 0 0 0 0 0 2 0 |
| 0 0 0 1 3 0 0 0 0 0 9 0 0 0 |
| 9 0 0 3 3 0 0 0 0 3 4 0 0 |
| 0 1 4 0 0 0 0 0 0 5 0 1 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 2 0 0 0 0 3 3 0 0 0 0 0 10 |
| 0 0 0 0 0 0 0 0 0 4 0 0 0 |
-----
Disponivel: 2 [ 3 delegacias colocadas]
Protegidos: 125 / 125
Solucao!
```

Instância 10 b

Instância	Alvo	Dimensões	Expansões	Gerações	Tempo Process.(ms)	Encontrou
10	126	[5]	0	81	0.00	0
10	126	[9]	0	25	0.00	0
10	126	[3]	0	121	1.00	0
10	126	[7]	0	49	0.00	0
10	126	[3, 7]	121	6050	68.11	0
10	126	[3, 5]	121	9922	82.93	0
10	126	[5, 7]	81	4050	46.07	0
10	126	[3, 3]	121	14762	136.19	0
10	126	[5, 5]	81	6642	55.12	0
10	126	[7, 7]	49	2450	28.02	0
10	126	[3, 3, 3]	14762	1786323	12661.10	0
10	126	[3, 5, 7]	9922	490171	6314.00	0
10	126	[5, 5, 7]	522	25750	388.38	1
Fim: 10	126		25780	2346396	19782.91	[[['(5, 6) 7'], '(3, 1) 5'], '(0, 6) 5']

```
Populacao:
[9, 0, 0, 0, 12]
[0, 0, 0, 1, 0]
[2, 0, 0, 2, 0]
[4, 0, 0, 0, 6]
[0, 0, 0, 0, 0]
Total: 36, Top-left: (0, 6) , Center: 3, 9, Raio: 2 , Solucao: True
Populacao:
[0, 0, 0, 0, 9]
[6, 9, 0, 0, 0]
[0, 0, 0, 0, 0]
[3, 0, 0, 0, 1]
[0, 0, 1, 3, 0]
Total: 32, Top-left: (3, 1) , Center: 6, 4, Raio: 2 , Solucao: True
Populacao:
[0, 1, 6, 10, 0, 1, 4]
[0, 0, 0, 0, 0, 2, 0]
[0, 0, 0, 9, 0, 0, 0]
[0, 0, 0, 3, 4, 0, 0]
[0, 0, 0, 5, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0]
[3, 0, 0, 0, 0, 0, 10]
Total: 59, Top-left: (5, 6) , Center: 9, 10, Raio: 3 , Solucao: True
```

	0	0	1	4	0	0	9	0	0	0	12	0	1	
	0	0	0	0	0	0	0	0	0	1	0	0	0	
	1	0	0	0	0	0	2	0	0	2	0	0	0	
	0	0	0	0	0	9	4	0	0	0	6	0	0	
	0	6	9	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	1	6	10	0	1	4	
	0	3	0	0	0	1	0	0	0	0	0	2	0	
	0	0	0	1	3	0	0	0	0	9	0	0	0	
	9	0	0	3	3	0	0	0	0	3	4	0	0	
	0	1	4	0	0	0	0	0	0	5	0	1	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	
	2	0	0	0	0	3	3	0	0	0	0	0	10	
	0	0	0	0	0	0	0	0	0	4	0	0	0	

Disponível: 1 [3 delegacias colocadas]
 Protegidos: 127 / 126
 Solucao!

ID 10

Proteger A - 125

128

25

66

37

		1	4			9				12		1
									1			
1						2			2			
					9	4				6		
	6	9										
							1	6	10		1	4
	3				1						2	
			1	3					9			
9			3	3					3	4		
	1	4							5		4	
2					3	3						10
									4			

Proteger B - 126 - sem solução

130

32

62

36

		1	4			9				12		1
									1			
1						2			2			
					9	4				6		
	6	9										
							1	6	10		1	4
	3				1						2	
			1	3					9			
9			3	3					3	4		
	1	4							5		4	
2					3	3						10
									4			

Função responsável pela procura cega em profundidade (Depth-first search)

```
def depth_first_search(matrix, dimensions, nivel, alvo):
    n_expansoes = 0
    n_geracoes = 0
    if len(dimensions) != 0:

        #, exclude_top_left, dimension):
        n = len(matrix)
        m = len(matrix[0])
        root_nodes = []

        # fazer uma copia da matriz atual
        original_matrix_copy = [row[:] for row in matrix]
        # fazer uma copia da matriz atual para trabalhar
        working_matrix = [row[:] for row in matrix]
        encontrou = False
        # Processa a dimensao da matriz
        i = dimensions[nivel]
        for r in range(n - i + 1):
            for c in range(m - i + 1):
                submatriz = [row[c:c + i] for row in matrix[r:r + i]]
                #verificar se tem algum elemento do pai
                #testava negativo para não haver sobreposicao
                #nunca há
                has_negative_one = any(-1 in row for row in submatriz)
                if (not has_negative_one):
                    total = sum(sum(row) for row in submatriz)

                    top_left = (r, c)
                    node = Estado(submatriz, total, top_left, False)
                    n_geracoes += 1
                    if total >= alvo:
                        encontrou = True
                        node.solucao = True
                    # colocar os valores da submatriz a zero para nao serem
                    # contabilizados nos outros niveis
                    for x in range(r, r + i):
                        for y in range(c, c + i):
                            # aqui se colocado a -1 passa a não permitir
sobreposição
                            working_matrix[x][y] = 0

                    #Agora posso criar o segundo nivel?
                    if ((len(dimensions)-1) != nivel) and (encontrou == False):
                        nivel += 1
                        n_expansoes += 1
                        #trata a expansao
                        node.children, a_expansoes, a_geracoes, encontrou =
depth_first_search(working_matrix, dimensions, nivel, alvo - total)
                        n_geracoes += a_geracoes
                        n_expansoes += a_expansoes
                        node.solucao = encontrou
                        nivel -= 1
                    if node.solucao:
                        #adiciona só nos parte da solucao
                        root_nodes.append(node)
                    if encontrou:
                        return root_nodes, n_expansoes, n_geracoes, True
                    working_matrix = [row[:] for row in matrix]

    return root_nodes, n_expansoes, n_geracoes, encontrou
```