

# Manual Utilizador

## Blokus Uno

Disciplina - Inteligencia Artificial

Turma - SW-01

Realizadores - Joana Guerreiro 202001733 | Andreia Novas - 201400498

## Índice

1. Acrónimos e Convenções usadas
2. Introdução
3. Instalação e utilização
4. Input/Output
5. Exemplo de aplicação

## Introdução

Este documento é um guia para o utilizar, da aplicação desenvolvida em Common Lisp, para a Unidade Curricular de Inteligência Artificial. Esta aplicação é um jogo de puzzle baseado no jogo conhecido como o Blokus, sendo o **Blokus Uno** uma versão do mesmo mais simplificado. No **Blokus Uno** existe um tabuleiro de 14 linhas e 14 colunas, existem 35 peças de três tipos, sendo que a peça C pode ser colocada em duas posições distintas, existe apenas um jogador no jogo e este termina quando todas as peças tiverem sido colocadas ou não for possível colocar mais peças respeitando as regras. Este jogo é iniciado com a escolha do tabuleira/problema e do algoritmo. O objectivo do jogo é preencher o tabuleiro com os 3 tipos de peças nos espaços mínimos existentes no problema, que se pode visualizar na escolha do tabuleiro. Nesta aplicação está desenvolvidos os algoritmos BFS, DFS e A\*, de procura em Espaços de Estados, para a resolução autónoma do jogos.

- **BFS(breadth-first Search)**: Algoritmo de procura em largura, explora todos os sucessores de um nó antes de passar para o nível seguinte, segue a ordem crescente dos nós da figura no topo da secção.
- **DFS(Depth-first Search)**: Algoritmo de procura em profundidade, explora sempre o primeiro sucessor de um nó, atingindo a máxima profundidade possível até encontrar uma solução, caso atinja o limite sem encontrar solução, percorre, utilizando os mesmos métodos, o sucessor mais próximo.
- **A\***: Algoritmo de procura informada utiliza uma função heurística para calcular o valor de um nó e apenas explora os nós com o menor valor heurístico, garantindo assim, a melhor solução possível.

## Instalação e utilização

Para a aplicação correr é necessário alterar o caminho da função diretório, pelo caminho onde o utilizador guardou a aplicação, no ficheiro projeto.lisp

- Exemplo windows : C:/Users/User/Instituto Politécnico de Setúbal/IA Project - Blocks - General/Blocks/bloks/parte1

O início da sua utilização é necessário iniciar o jogo, escolher um problema que será apresentado no menu , também será apresentado um 3 tipos algoritmos , para a resolução do problema.

## Input/Output

```
Bem Vindo - Blokus Uno
1 - Jogar
2 - Sair
```

No início da aplicação será apresentado :

[illegible]

2 -	A B C D E F G H I J K L M N
1	(0 0 0 0 0 0 0 2 2 2 2 2 2)
2	(0 0 0 0 0 0 0 2 2 2 2 2 2)
3	(0 0 0 0 0 0 0 2 2 2 2 2 2)
4	(0 0 0 0 0 0 0 2 2 2 2 2 2)
5	(0 0 0 0 0 0 0 2 2 2 2 2 2)
6	(0 0 0 0 0 0 0 2 2 2 2 2 2)
7	(0 0 0 0 0 0 0 2 2 2 2 2 2)
8	(2 2 2 2 2 2 2 2 2 2 2 2 2)
9	(2 2 2 2 2 2 2 2 2 2 2 2 2)
10	(2 2 2 2 2 2 2 2 2 2 2 2 2)
11	(2 2 2 2 2 2 2 2 2 2 2 2 2)
12	(2 2 2 2 2 2 2 2 2 2 2 2 2)
13	(2 2 2 2 2 2 2 2 2 2 2 2 2)
14	(2 2 2 2 2 2 2 2 2 2 2 2 2)

[illegible][illegible]

```

5(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
6(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
7(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
8(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
9(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
10(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
11(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
12(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
13(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
14(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
Preencher pelo menos 36 casas para ganhar

```

```

5 -   A B C D E F G H I J K L M N
      1(0 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
      2(2 0 2 0 0 0 0 0 0 0 2 0 0 0 2)
      3(2 0 0 2 0 0 0 0 0 0 0 2 0 0 2)
      4(2 0 0 0 2 0 0 0 0 0 0 0 2 0 2)
      5(2 0 0 0 0 2 0 0 0 0 0 0 0 2 2)
      6(2 0 0 0 0 0 2 0 0 0 0 0 0 0 2)
      7(2 0 0 0 0 0 0 2 0 0 0 0 0 0 2)
      8(2 0 0 0 0 0 0 0 2 0 0 0 0 0 2)
      9(2 0 0 0 0 0 0 0 0 2 0 0 0 0 2)
     10(2 0 2 0 0 0 0 0 0 0 0 2 0 0 2)
     11(2 2 0 0 0 0 0 0 0 0 0 0 2 0 2)
     12(2 0 0 0 2 0 0 0 0 0 0 0 0 2 2)
     13(2 0 0 0 0 2 0 0 0 0 0 0 0 0 2)
     14(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
Preencher pelo menos 44 casas para ganhar

```

```

6 -   A B C D E F G H I J K L M N
      1(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      2(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      3(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      4(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      5(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      6(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      7(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      8(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
      9(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
     10(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
     11(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
     12(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
     13(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
     14(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
Preencher pelo menos 72 casas para ganhar

```

Terá de escolher um algoritmo

> Escolha um algoritmo para aplicar na resolucao do problema:

- 1 - BFS
- 2 - DFS
- 3 - A\*

Terá de inserir uma máxima profundidade se escolher o dfs

```
> Introduza a profundidade maximo do algortmo
```

Terá de escolher uma heuristica se for o a\*

```
> Escolha uma heuristica:
1 - Heuristica default (fornecida no enunciado)
2 - Heuristica custom (criado pelo grupo)
```

## Exemplo de aplicação

No exemplo de aplicação destes problemas, acrescentamos um problema novo que é referido no enunciado e escolhemos o algoritmo a\* e a nova heuristica criada.

```
Bem Vindo - Blokus Uno
```

```
1 - Jogar
2 - Sair
```

```
1|
```

É mostrado todas os problemas existentes.

```
7 -  A B C D E F G H I J K L M N
    1(0 0 0 0 2 2 2 2 2 2 2 2 2 2)
    2(0 0 0 0 2 2 2 2 2 2 2 2 2 2)
    3(0 0 0 0 0 0 2 2 2 2 2 2 2 2)
    4(0 0 0 0 0 0 2 2 2 2 2 2 2 2)
    5(0 0 0 0 0 0 0 0 2 2 2 2 2 2)
    6(0 0 0 0 0 0 0 0 2 2 2 2 2 2)
    7(0 0 0 0 0 0 0 0 0 0 2 2 2 2)
    8(2 0 0 0 0 0 0 0 0 0 2 2 2 2)
    9(2 2 0 0 0 0 0 0 0 0 0 0 2 2)
   10(2 2 2 0 0 0 0 0 0 0 0 0 2 2)
   11(2 2 2 2 0 0 0 0 0 0 0 0 0 0)
   12(2 2 2 2 2 0 0 0 0 0 0 0 0 0)
   13(2 2 2 2 2 2 0 0 0 0 0 0 0 0)
   14(2 2 2 2 2 2 2 0 0 0 0 0 0 0)
    Preencher pelo menos 49 casas para ganhar
```

```
7|
```

Neste exemplo escolhi o 7 , novo problema adicionado.

> Escolha um algoritmo para aplicar na resolucao do problema:

- 1 - BFS
- 2 - DFS
- 3 - A\*

3

Escolhemos o problema a \* .

> Escolha uma heuristica:

- 1 - Heuristica default (fornecida no enunciado)
  - 2 - Heuristica custom (criado pelo grupo)
- 2|

Escolhemos a heuristica que criamos.

De seguida passo a apresentar as caraterísticas dos problemas.

> Caracteristicas:

- Algoritmo: A\*
- Heuristica: HEURISTICA-CUSTOM
- Profundidade: 0
- Problema:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	0	0	0	2	2	2	2	2	2	2	2	2	2
2	0	0	0	0	2	2	2	2	2	2	2	2	2	2
3	0	0	0	0	0	0	2	2	2	2	2	2	2	2
4	0	0	0	0	0	0	2	2	2	2	2	2	2	2
5	0	0	0	0	0	0	0	0	2	2	2	2	2	2
6	0	0	0	0	0	0	0	0	2	2	2	2	2	2
7	0	0	0	0	0	0	0	0	0	2	2	2	2	2
8	2	0	0	0	0	0	0	0	0	0	2	2	2	2
9	2	2	0	0	0	0	0	0	0	0	0	2	2	2
10	2	2	2	0	0	0	0	0	0	0	0	0	2	2
11	2	2	2	2	0	0	0	0	0	0	0	0	0	0
12	2	2	2	2	2	0	0	0	0	0	0	0	0	0
13	2	2	2	2	2	2	0	0	0	0	0	0	0	0
14	2	2	2	2	2	2	2	0	0	0	0	0	0	0

Minimo Casas: 49  
Casas Preenchidas: 0  
Peca Quadrado Pequeno: 10  
Peca Quadrado: 10  
Peca S: 15

Por fim os resultados , o primeiro a ser apresentado é o no solução.



> Resultados:

- Nos gerados: 382
- Nos expandidos: 295
- Penetrancia: 0.041884818
- Fator de ramificacao: 1.3299217
- Tempo de execucao: 2.69 segundo(s)
- Profundidade da solucao: 16
- Solucao:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	1	1	0	0	2	2	2	2)
8	(2	0	1	1	0	0	1	1	0	0	2	2	2	2)
9	(2	2	0	0	1	1	0	0	1	1	0	0	2	2)
10	(2	2	2	0	1	1	0	0	1	1	0	0	2	2)
11	(2	2	2	2	0	0	1	1	0	0	1	1	0	1)
12	(2	2	2	2	2	0	1	1	0	1	1	0	1	0)
13	(2	2	2	2	2	0	0	1	0	0	0	1	1)	
14	(2	2	2	2	2	2	1	0	1	0	1	0	1)	

Minimo Casas: 49

Casas Preenchidas: 49

Peca Quadrado Pequeno: 5

Peca Quadrado: 1

Peca S: 13

Minimo Casas: 49

Casas Preenchidas: 47

Peca Quadrado Pequeno: 7

Peca Quadrado: 1

Peca S: 13

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	1	1	0	0	2	2	2	2)
8	(2	0	1	1	0	0	1	1	0	0	2	2	2	2)
9	(2	2	0	0	1	1	0	0	1	1	0	0	2	2)
10	(2	2	2	0	1	1	0	0	1	1	0	0	2	2)
11	(2	2	2	2	0	0	1	1	0	0	1	1	0	1)
12	(2	2	2	2	2	0	1	1	0	1	1	0	1	0)
13	(2	2	2	2	2	0	0	1	0	0	0	1	1)	
14	(2	2	2	2	2	2	0	0	0	0	0	0	1)	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	1	1	0	0	2	2	2	2)
8	(2	0	1	1	0	0	1	1	0	0	2	2	2	2)
9	(2	2	0	0	1	1	0	0	1	1	0	0	2	2)
10	(2	2	2	0	1	1	0	0	1	1	0	0	2	2)
11	(2	2	2	2	0	0	1	1	0	0	1	1	0	1)
12	(2	2	2	2	2	0	1	1	0	1	1	0	1	0)
13	(2	2	2	2	2	0	0	1	0	0	0	1	1)	
14	(2	2	2	2	2	2	1	0	1	0	0	0	1)	

Minimo Casas: 49

Casas Preenchidas: 48

Peca Quadrado Pequeno: 6

Peca Quadrado: 1

Peca S: 13

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	1	1	0	0	2	2	2	2)
8	(2	0	1	1	0	0	1	1	0	0	2	2	2	2)
9	(2	2	0	0	1	1	0	0	1	1	0	0	2	2)
10	(2	2	2	0	1	1	0	0	1	1	0	0	2	2)
11	(2	2	2	2	0	0	1	1	0	0	1	1	0	1)
12	(2	2	2	2	2	0	1	1	0	1	1	0	1	0)
13	(2	2	2	2	2	0	0	1	0	0	0	1	1)	
14	(2	2	2	2	2	2	0	0	1	0	0	0	1)	





	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	1	0	0	2	2	2	2	2	2	2	2	2	2
2	1	1	0	0	2	2	2	2	2	2	2	2	2	2
3	0	0	1	1	0	0	2	2	2	2	2	2	2	2
4	0	0	1	1	0	0	2	2	2	2	2	2	2	2
5	1	1	0	0	1	1	0	0	2	2	2	2	2	2
6	1	1	0	0	1	1	0	0	2	2	2	2	2	2
7	0	0	1	1	0	0	1	1	0	0	2	2	2	2
8	2	0	1	1	0	0	1	1	0	0	2	2	2	2
9	2	2	0	0	1	1	0	0	1	1	0	0	2	2
10	2	2	2	0	1	1	0	0	1	1	0	0	2	2
11	2	2	2	2	0	0	0	0	0	0	0	0	0	0
12	2	2	2	2	2	0	0	0	0	0	0	0	0	0
13	2	2	2	2	2	2	0	0	0	0	0	0	0	0
14	2	2	2	2	2	2	2	0	0	0	0	0	0	0

Minimo Casas: 49  
 Casas Preenchidas: 32  
 Peca Quadrado Pequeno: 10  
 Peca Quadrado: 2  
 Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	1	0	0	2	2	2	2	2	2	2	2	2	2
2	1	1	0	0	2	2	2	2	2	2	2	2	2	2
3	0	0	1	1	0	0	2	2	2	2	2	2	2	2
4	0	0	1	1	0	0	2	2	2	2	2	2	2	2
5	1	1	0	0	1	1	0	0	2	2	2	2	2	2
6	1	1	0	0	1	1	0	0	2	2	2	2	2	2
7	0	0	1	1	0	0	1	1	0	0	2	2	2	2
8	2	0	1	1	0	0	1	1	0	0	2	2	2	2
9	2	2	0	0	1	1	0	0	0	0	0	0	2	2
10	2	2	2	0	1	1	0	0	0	0	0	0	2	2
11	2	2	2	2	0	0	0	0	0	0	0	0	0	0
12	2	2	2	2	2	0	0	0	0	0	0	0	0	0
13	2	2	2	2	2	2	0	0	0	0	0	0	0	0
14	2	2	2	2	2	2	2	0	0	0	0	0	0	0

Minimo Casas: 49

---

Minimo Casas: 49  
 Casas Preenchidas: 28  
 Peca Quadrado Pequeno: 10  
 Peca Quadrado: 3  
 Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	1	1	0	0	2	2	2	2)
8	(2	0	1	1	0	0	1	1	0	0	2	2	2	2)
9	(2	2	0	0	0	0	0	0	0	0	0	0	2	2)
10	(2	2	2	0	0	0	0	0	0	0	0	0	2	2)
11	(2	2	2	2	0	0	0	0	0	0	0	0	0	0)
12	(2	2	2	2	2	0	0	0	0	0	0	0	0	0)
13	(2	2	2	2	2	2	0	0	0	0	0	0	0	0)
14	(2	2	2	2	2	2	2	0	0	0	0	0	0	0)

Minimo Casas: 49  
 Casas Preenchidas: 24  
 Peca Quadrado Pequeno: 10  
 Peca Quadrado: 4  
 Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
4	(0	0	1	1	0	0	2	2	2	2	2	2	2	2)
5	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
6	(1	1	0	0	1	1	0	0	2	2	2	2	2	2)
7	(0	0	1	1	0	0	0	0	0	0	2	2	2	2)
8	(2	0	1	1	0	0	0	0	0	0	2	2	2	2)
9	(2	2	0	0	0	0	0	0	0	0	0	0	2	2)

---



5(1 1 0 0 0 0 0 0 2 2 2 2 2 2)  
 6(1 1 0 0 0 0 0 0 2 2 2 2 2 2)  
 7(0 0 0 0 0 0 0 0 0 0 2 2 2 2)  
 8(2 0 0 0 0 0 0 0 0 0 2 2 2 2)  
 9(2 2 0 0 0 0 0 0 0 0 0 0 2 2)  
 10(2 2 2 0 0 0 0 0 0 0 0 0 2 2)  
 11(2 2 2 2 0 0 0 0 0 0 0 0 0 0)  
 12(2 2 2 2 2 0 0 0 0 0 0 0 0 0)  
 13(2 2 2 2 2 2 0 0 0 0 0 0 0 0)  
 14(2 2 2 2 2 2 2 0 0 0 0 0 0 0)

Minimo Casas: 49

Casas Preenchidas: 12

Peca Quadrado Pequeno: 10

Peca Quadrado: 7

Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	1	0	0	2	2	2	2	2	2	2	2	2	2
2	1	1	0	0	2	2	2	2	2	2	2	2	2	2
3	0	0	1	1	0	0	2	2	2	2	2	2	2	2
4	0	0	1	1	0	0	2	2	2	2	2	2	2	2
5	0	0	0	0	0	0	0	0	2	2	2	2	2	2
6	0	0	0	0	0	0	0	0	2	2	2	2	2	2
7	0	0	0	0	0	0	0	0	0	0	2	2	2	2
8	2	0	0	0	0	0	0	0	0	0	2	2	2	2
9	2	2	0	0	0	0	0	0	0	0	0	0	2	2
10	2	2	2	0	0	0	0	0	0	0	0	0	2	2
11	2	2	2	2	0	0	0	0	0	0	0	0	0	0
12	2	2	2	2	2	0	0	0	0	0	0	0	0	0
13	2	2	2	2	2	2	0	0	0	0	0	0	0	0
14	2	2	2	2	2	2	2	0	0	0	0	0	0	0

Minimo Casas: 49

Casas Preenchidas: 8

Peca Quadrado Pequeno: 10

Peca Quadrado: 8

Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
2	(1	1	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	0	0	0	0	2	2	2	2	2	2	2	2)
4	(0	0	0	0	0	0	2	2	2	2	2	2	2	2)
5	(0	0	0	0	0	0	0	0	2	2	2	2	2	2)
6	(0	0	0	0	0	0	0	0	2	2	2	2	2	2)
7	(0	0	0	0	0	0	0	0	0	0	2	2	2	2)
8	(2	0	0	0	0	0	0	0	0	0	2	2	2	2)
9	(2	2	0	0	0	0	0	0	0	0	0	0	2	2)
10	(2	2	2	0	0	0	0	0	0	0	0	0	2	2)
11	(2	2	2	2	0	0	0	0	0	0	0	0	0	0)
12	(2	2	2	2	2	0	0	0	0	0	0	0	0	0)
13	(2	2	2	2	2	2	0	0	0	0	0	0	0	0)
14	(2	2	2	2	2	2	2	0	0	0	0	0	0	0)

Minimo Casas: 49  
 Casas Preenchidas: 4  
 Peca Quadrado Pequeno: 10  
 Peca Quadrado: 9  
 Peca S: 15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
2	(0	0	0	0	2	2	2	2	2	2	2	2	2	2)
3	(0	0	0	0	0	0	2	2	2	2	2	2	2	2)
4	(0	0	0	0	0	0	2	2	2	2	2	2	2	2)
5	(0	0	0	0	0	0	0	0	2	2	2	2	2	2)
6	(0	0	0	0	0	0	0	0	2	2	2	2	2	2)
7	(0	0	0	0	0	0	0	0	0	0	2	2	2	2)
8	(2	0	0	0	0	0	0	0	0	0	2	2	2	2)
9	(2	2	0	0	0	0	0	0	0	0	0	0	2	2)
10	(2	2	2	0	0	0	0	0	0	0	0	0	2	2)
11	(2	2	2	2	0	0	0	0	0	0	0	0	0	0)
12	(2	2	2	2	2	0	0	0	0	0	0	0	0	0)
13	(2	2	2	2	2	2	0	0	0	0	0	0	0	0)
14	(2	2	2	2	2	2	2	0	0	0	0	0	0	0)

Minimo Casas: 49

Casas Preenchidas: 0  
 Peca Quadrado Pequeno: 10  
 Peca Quadrado: 10  
 Peca S: 15

NIL