

# LightUP

Uma competição da disciplina FLIA

Bruno César Ribas

Faculdade de Ciências e Tecnologias em Engenharia - FCTE  
Universidade de Brasília

January 7, 2025

# Overview

---

## 1. LightUP

## 2. Competição

## 3. Soluções

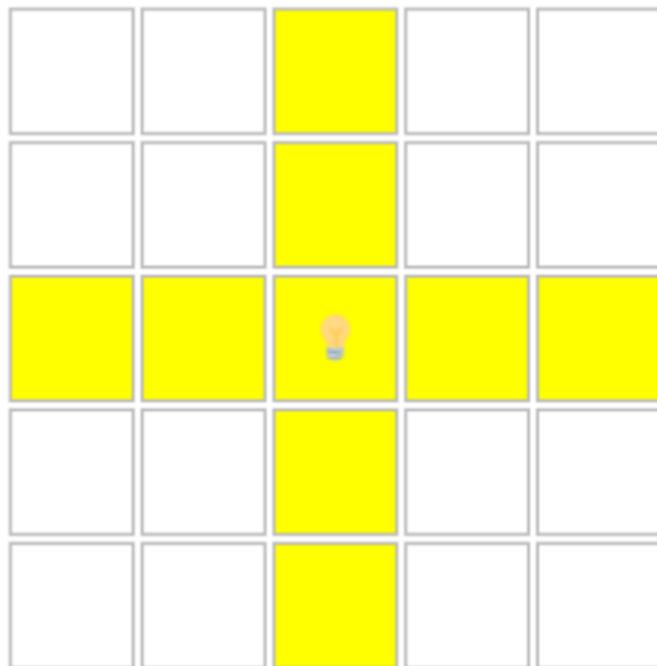
3.1 PDDL

3.2 pseudo-Boolean

- Também conhecido como AKARI
- Publicado pela NIKOLI - <https://www.nikoli.co.jp/en/>
- Objetivo é colocar lâmpadas no GRID e deixar todas as células iluminadas
- Cada lâmpada ilumina a linha e a coluna da posição em que foi colocada
- Duas, ou mais lâmpadas, não podem se iluminar
- Existem Paredes
- Existem células numeradas, que exigem uma quantidade de lâmpadas adjacentes
- Trivial validar as regras, difícil de resolver

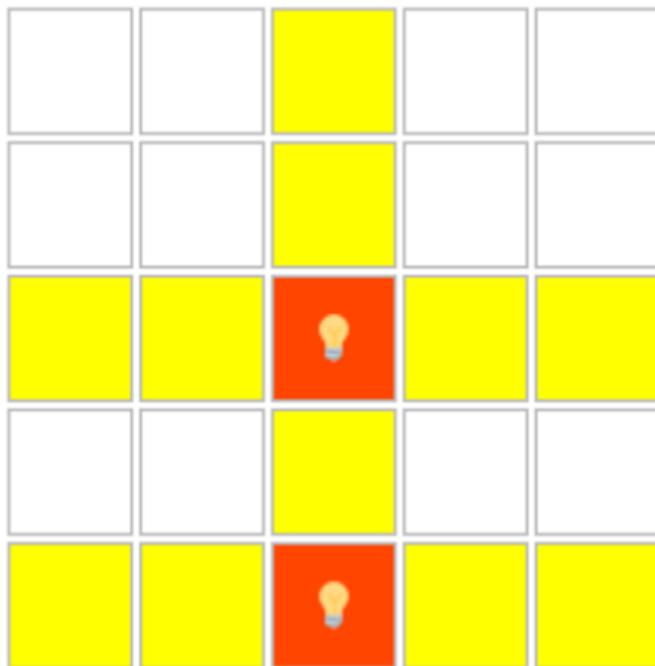
# Regras do Jogo

---



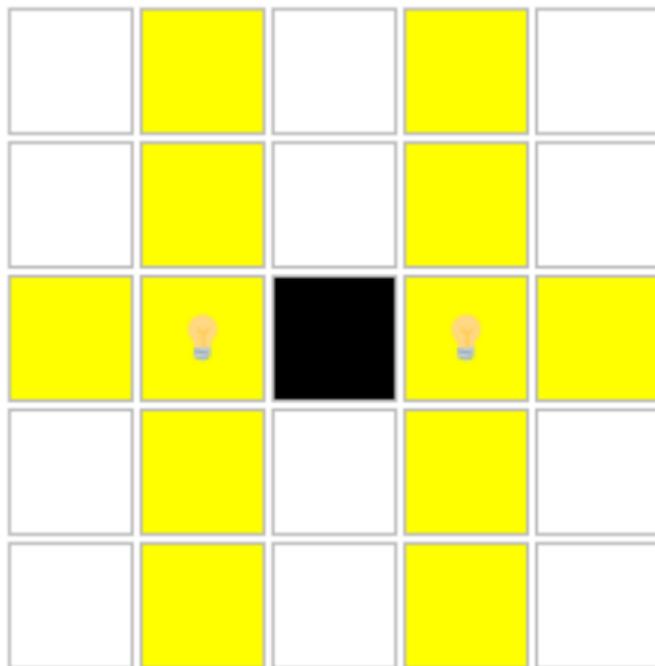
# Regras do Jogo

---



# Regras do Jogo

---



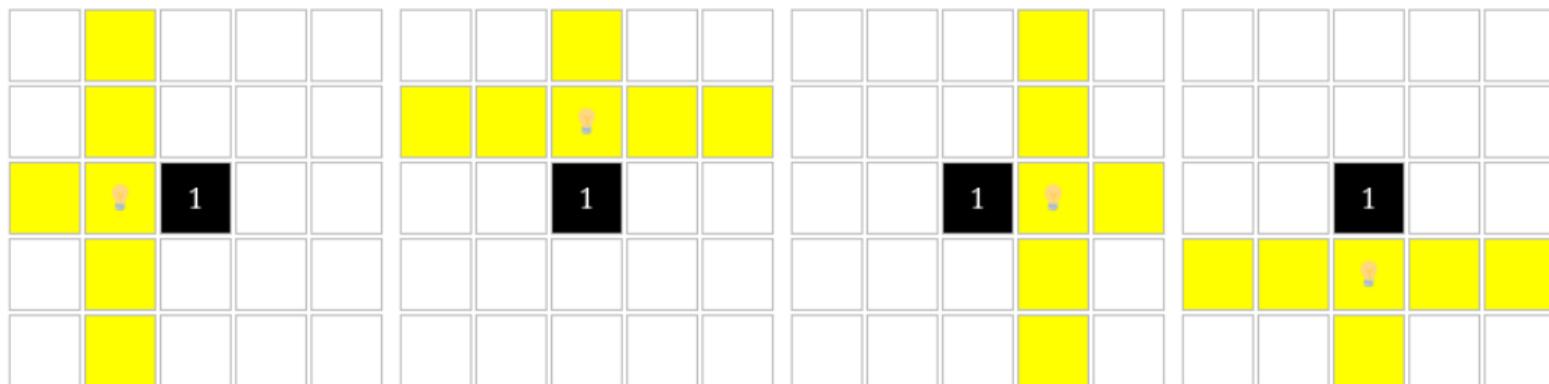
# Regras do Jogo

---

		0		

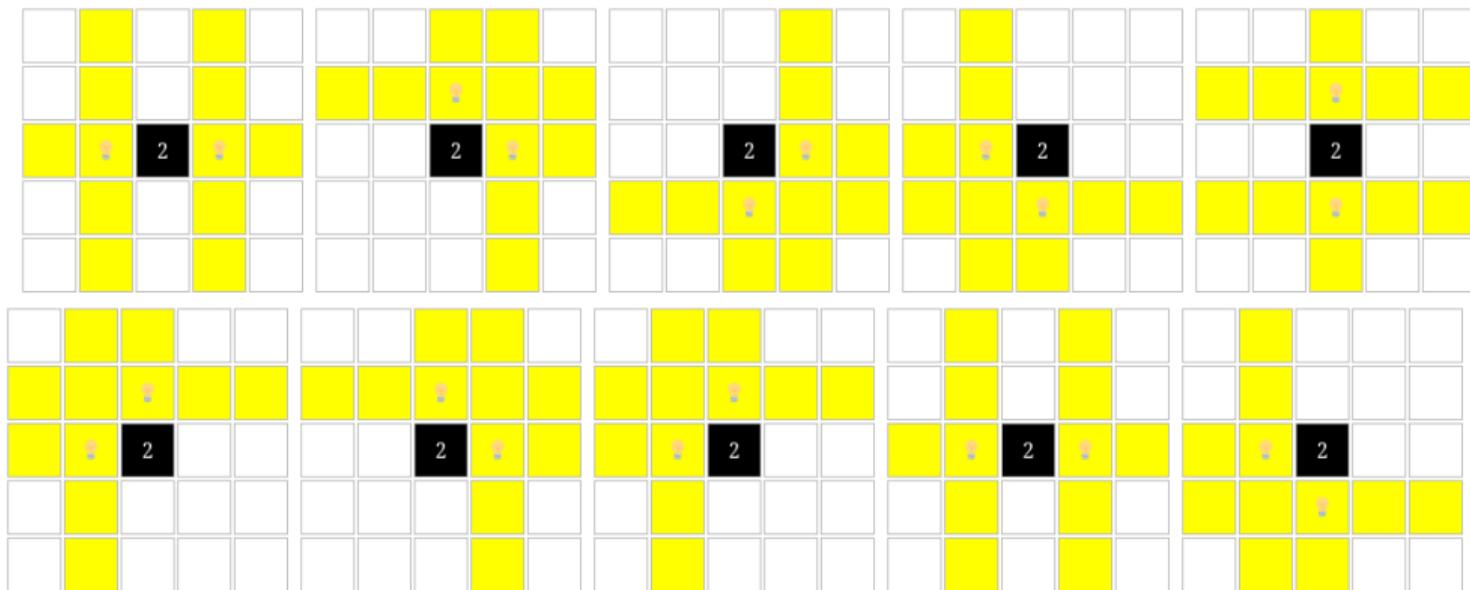
# Regras do Jogo

---



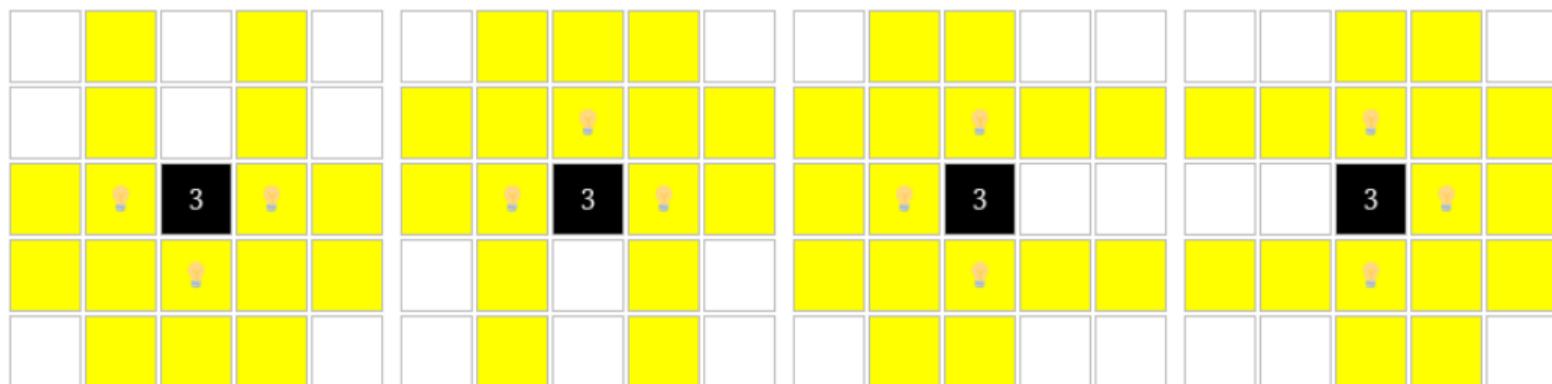
# Regras do Jogo

---



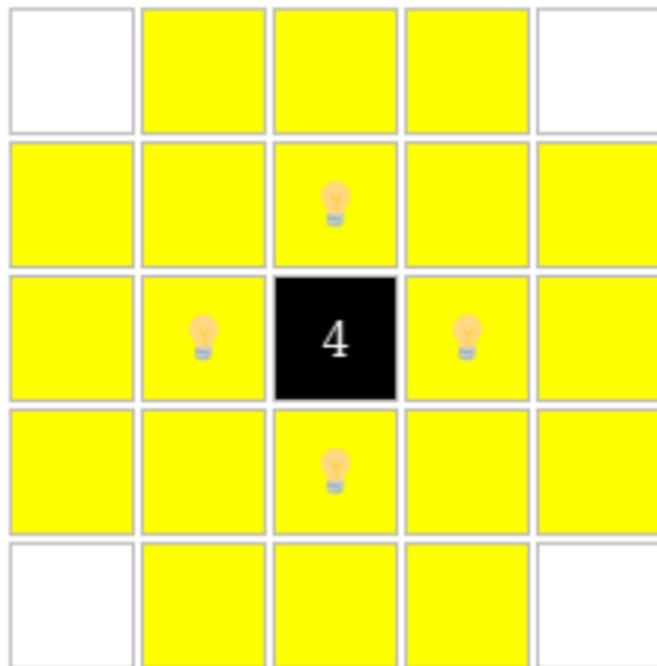
# Regras do Jogo

---



# Regras do Jogo

---



# A competição

---

## AGILE

Objetivo: Achar uma solução o mais rápido possível

Score: Se Tempo de Execucao  $> 1$ ;  $1 - \log(\text{TEMPODEEXECUCAO})/\log(30)$ ; Senão 1

TimeLimit: 30 segundos por mapa

## SATISFICING

Objetivo: Achar uma solução intermediária, menor quantidade de lâmpadas dentro do tempo limite.

Score:  $C^*/C$ , sendo  $C^*$  referência, e  $C$  solução

TimeLimit: 180 segundos

## Examples

Objetivo: Encontrar uma solução com a menor quantidade de lâmpadas

Score: 1 ponto por mapa

TimeLimit: 180 segundos

# A competição

---

## **AGILE**

1. Core i7-9700 @ 3.00GHz
2. 16GB de memória
3. Turbo Boost Desligado
4. 2 máquinas idênticas (pos1, pos2)

## **SAT e OPT**

1. Core i7-8700 @ 3.20GHz
2. 16GB de memória
3. Turbo Boost LIGADO
4. 4 máquinas idênticas(cm1, cm2, cm3, cm4)

- Principais aspectos durante o desenvolvimento
- Propagação da Luz nas linhas e colunas dentro do Domínio é lento
- FastDownward não dá conta da maioria dos domínios criados
- Madagascar é o planejador mais eficiente, no entanto é necessário minerar alguns parâmetros

## PDDL - Solução A

---

- Uma ação por posição do mapa
- Predicados de Lampada por posição e predicado de Aceso por posição
- Ações não possuem parâmetros
- A pre-condição é que nenhuma lâmpada da linha e coluna tenha sido colocada, e que esta própria lâmpada não tenha sido colocada
- A pre-condição é montada via pré-processamento da entrada (há um custo, linguagens lentas impactam o desempenho)
- Como efeito: ativar predicado de que esta lâmpada está acesa e marcar todas as posições de linhas e colunas como acesas
- No problema, o :GOAL é que todas as posições estejam acesas
- O :INIT é vazio

## E para as células N?

- Criar um contador para cada célula N
- A pré-condição de cada ação de célula adjacente à célula N verifica se o contador não chegou no máximo
- O efeito é uma cadeia de WHENS que verifica se o contador está em um valor e avança para o próximo
- Estratégia dos WHENS para evitar parâmetros nas ações, causa domínios maiores
- no :GOAL adicionar que os contadores devem estar no valor máximo
- Esta estratégia limita o paralelismo de aplicação de ações do Madagascar

# PDDL - Solução A

```
function criaacao()
{
  for imp in ${NUMBER[impacto,$linha,$coluna]}; do
    preimpacto+="(not (contador-em $imp cont${NUMBER[cont,$imp]}))"
    #preimpacto+="(or"
    for i in 0 1 2 3; do
      (( i >= ${NUMBER[cont,$imp]} )) && continue
      #preimpacto+=" (contador-em $imp cont$i)"
      eff+="(when (contador-em $imp cont$i) (and (not (contador-em $imp cont$i))"
      (contador-em $imp cont${(i+1)}))"
    done
    #preimpacto+=")"
  done
  DOMINIOSTR+="
(:action BULB_${linha}_${coluna}
 :parameters ()
 :precondition (and (not (aceso P_${linha}_${coluna})) $preimpacto )
 :effect (and $eff (tem-lampada P_${linha}_${coluna}))
)"
  buscalinhacoluna $linha $coluna DOMINIOSTR
  DOMINIOSTR+="))"
}
```

# PDDL - Solução A

```
GOAL+="(forall (?p - posicao) (aceso ?p))"
for Q in $QNUM; do
  POSITIVONINICIO+="(contador-em $Q cont0) (contador-max $Q cont${NUMBER[cont,
$Q]})"
  GOAL+="(contador-em $Q cont${NUMBER[cont,$Q]})"
done
POSITIVONINICIO+="(nextc cont0 cont1) (nextc cont1 cont2) (nextc cont2 cont3)"

echo "$DOMINIOSTR" > $DOMAINFILE
cat > $PROBLEMFILE <<EOF
(define (problem problema)
  (:domain akari)
  (:init $POSITIVONINICIO)
  (:goal (and $GOAL)))
EOF
```

# pseudo-Boolean

---