

Departamento de Ciência da Computação  
D P I

## A Ciência de Fill a Pix

Wallace Ferancini Rosa - 92545  
Caio Oliveira Helmut Henrici - 92558

# 1. Introdução

Fill a pix se trata de um jogo, um desafio lógico, que consiste em formar uma imagem a partir de números em uma matriz. Tais números indicam a quantidade de quadrados que devem ser pintados no espaço 3x3 gerado, tendo o número indicante da quantidade como centro. Por exemplo, caso o número seja 9, significa que todos os quadrados no espaço 3x3 gerado pelo quadrado com número 9 devem ser pintados. A partir disso podemos gerar diversas técnicas de indução para descobrir quais quadrados devem, ou não, ser pintados para resultar na imagem desejada.

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

## 2. Métodos Utilizados

### Indução por casos base

Esse método recebe as dimensões da matriz desejada e a matriz contendo os números indicando quantos quadrados devem ser pintados no espaço 3x3 e gera uma nova matriz com a solução desejada como saída. Cada quadrado que deve ser pintado será representado na nova matriz como 1, os que não devem ser pintados serão representados como 0 e os que não sofreram interação serão representados por -1. O algoritmo começa checando a existência de casos básicos, onde são evidentes quais quadrados que devem ser pintados:

9, onde todos os quadrados a sua volta devem ser pintado;

0, onde nenhum quadrado a sua volta pode ser pintado;

4, onde houver um quatro em algum vértice todos os quadrados a sua volta devem ser pintados;

6, onde houver um seis em alguma linha ou coluna extrema (próximo as bordas) todos os quadrados a sua volta devem ser pintados;

A partir da solução de tais casos começaremos a varrer a matriz em busca da solução de espaços próximos aos já solucionados através de um while, tendo como condição de continuidade um bool, o qual será false quando não existirem espaços inalterados (-1) na nova matriz. Ao checar o número de quadrados livres no espaço 3x3 de cada coordenada e compará-lo ao número indicado na mesma é possível descobrir se tal espaço pode ser completado. Caso positivo, os quadrados possíveis de serem preenchidos, diferente de -1, não pintados em tal espaço serão preenchidos por 1 (pintados). Em seguida será checado a quantidade de quadrados pintados em torno de cada posição, caso essa quantidade seja igual ao número indicado na coordenada, significa que tal espaço já está satisfeito, assim preenchendo os espaços restantes com 0, evitando-os de serem pintados. Assim o programa rodará até ter alterado todas as posições da matriz até poder sair do while de comparações. Em seguida será imprimida a matriz resultado e desalocado qualquer espaço de memória usado pelas matrizes. Contudo, é necessário ressaltar que tal abordagem nem tem capacidade para gerar soluções onde tais casos bases não são aplicados ou não são suficientes para completar o resultado desejado, gerando a incapacidade de resolver problemas de elevada complexidade.

## Backtring por possibilidade

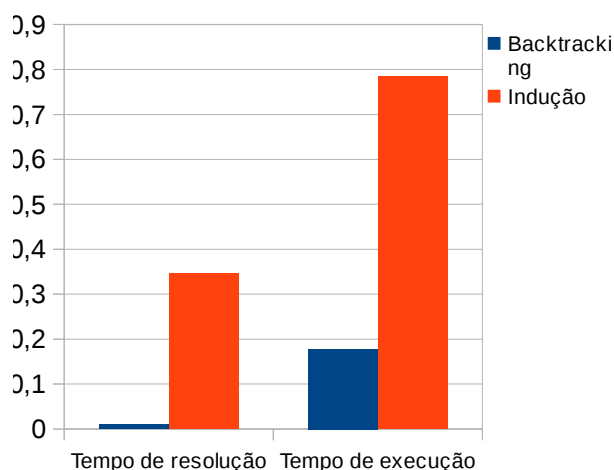
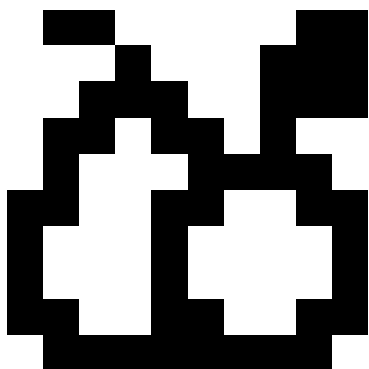
Esse método, igualmente ao primeiro, recebe as dimensões da matriz desejada e a matriz contendo os números indicando quantos quadrados devem ser pintados no espaço 3x3 e gera uma nova matriz com a solução desejada como saída. Cada quadrado que deve ser pintado sera representado na nova matriz como 1, os que não devem ser pintados serão representados como 0 e os que não sofreram interação serão representados por -1.

Nesse algoritmo começamos varrendo a matriz, pintando uma combinação de quadrados pretos no espaço 3x3 indicado pelo índice do quadrado centro. Em seguida é verificado se essa combinação entra em conflito com alguma posição ao seu redor, caso não haja conflito é prosseguido para a próxima posição da matriz. Caso haja conflito, a combinação é marcada como impossível de ser pintada, assim marcada como 0. Caso o número de quadrados marcados como não pintáveis (0) no espaço 3x3 seja maior que o número indicado pelo quadrado centro subtraído pela a quantidade de quadrados não pintáveis no espaço 3x3, significa que essa combinação não é possível de ser realizada. Assim é combinação é descartada e a função para testar combinações é chamada novamente começando uma nova sequência de combinações até que o problema seja resolvido. É importante ressaltar que tal método pode gerar imagens finais diferentes das desejadas por certas combinações fugirem de serem impossíveis ao se juntarem com outras combinações inesperadas, podendo resultar em parte da imagem final diferentes do resultado esperado, as chances são mais prováveis de acordo com o aumento do nível de dificuldade.

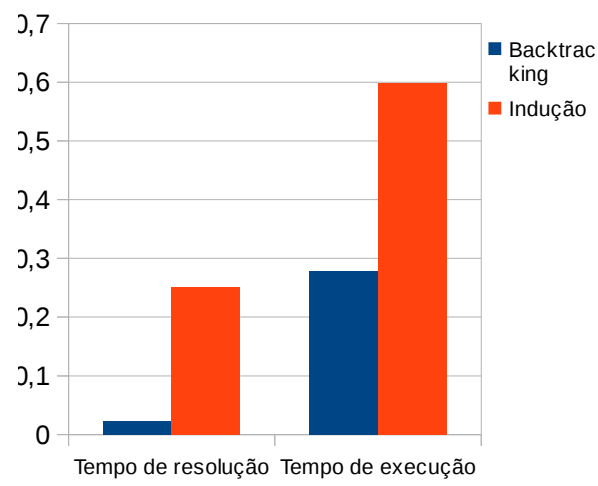
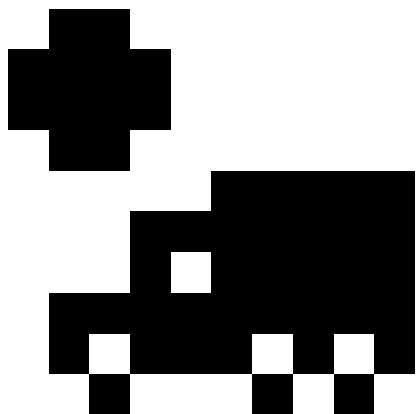
## 3. Testes

Os algoritmos serão analisados em função de seu tempo (em milisegundos) de execução e no nível de dificuldade do quebra cabeças 10x10. Serão usados caso simples e avançados para matrizes 10x10, para as 15x15 e 20x20 a própria quantidade de combinações possíveis faz com que o problema apresente grade complexidade. Os testes devem comparar a eficiência dos métodos utilizados e sua capacidade de resolver casos com maior nível de complexidade.

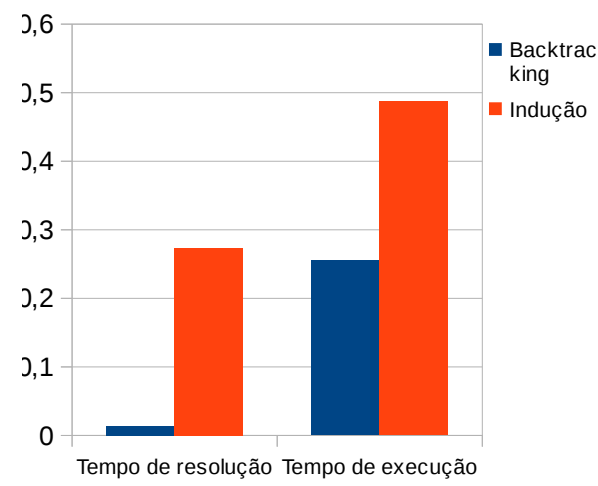
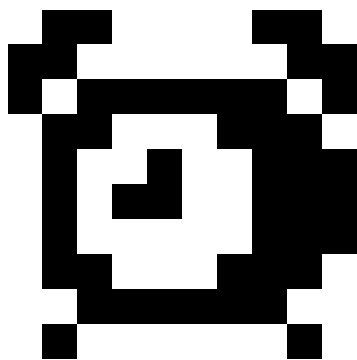
Fácil - 10x10



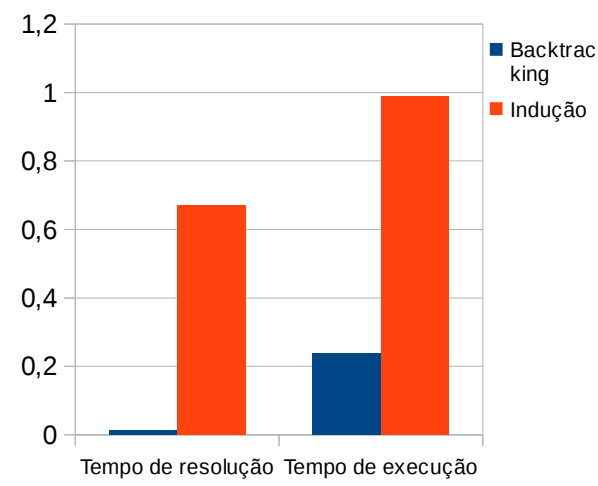
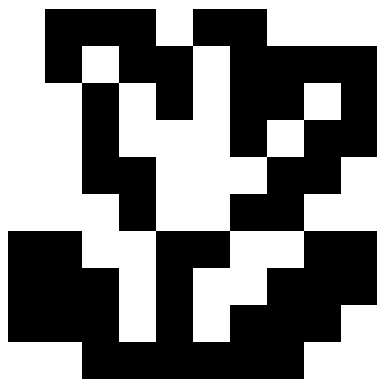
Facíl - 10x10



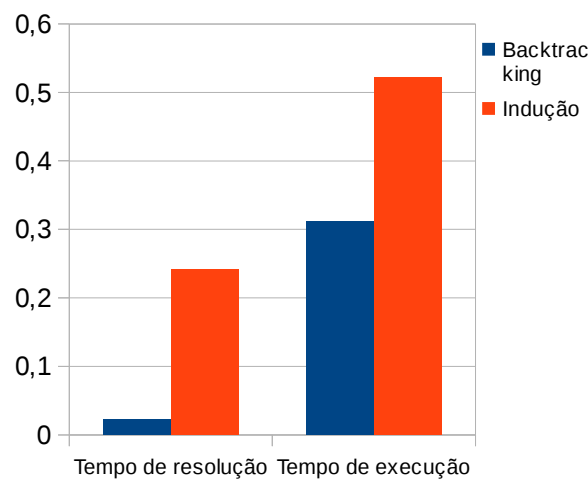
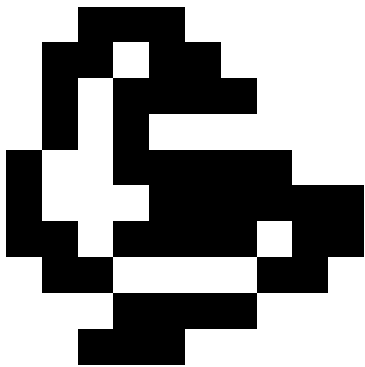
Facíl - 10x10



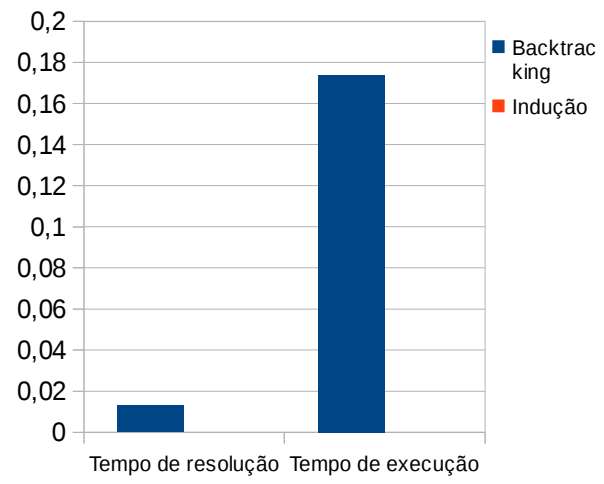
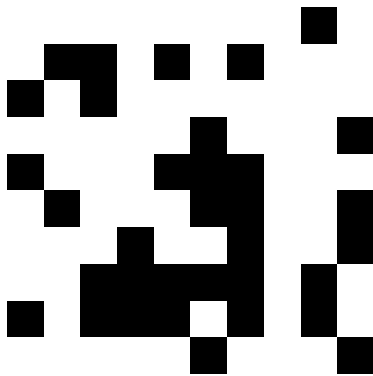
Facíl - 10x10



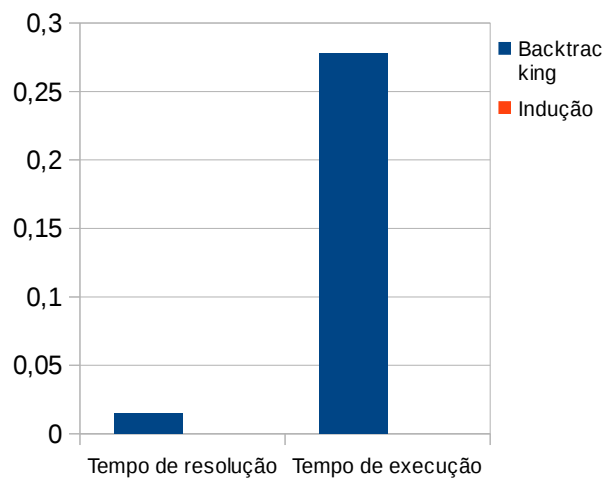
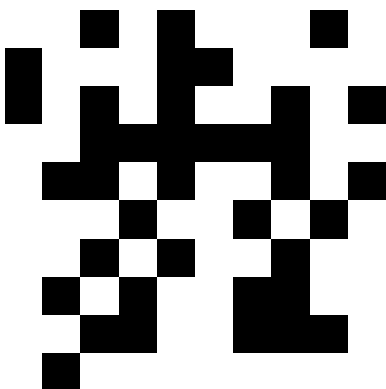
Fácil - 10x10



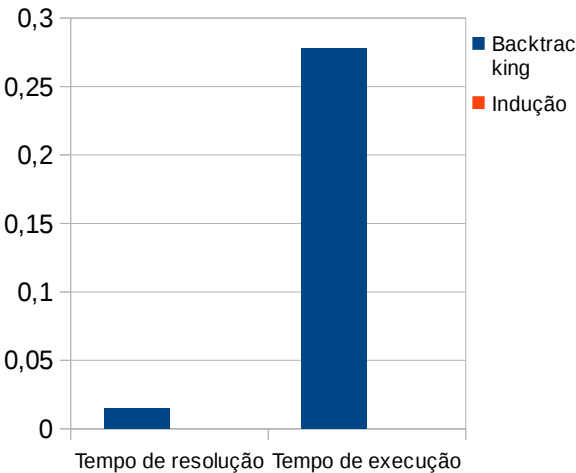
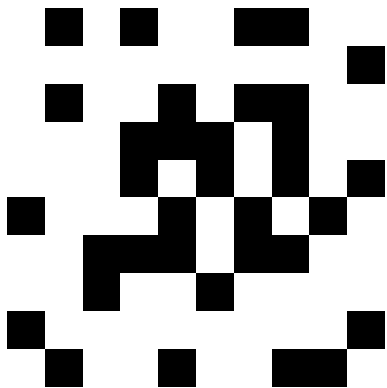
Difícil - 10x10



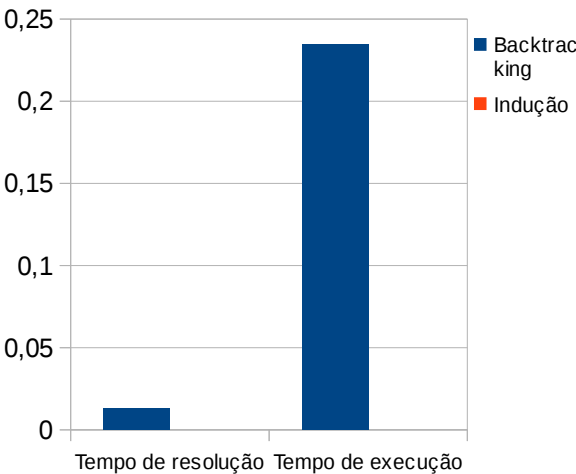
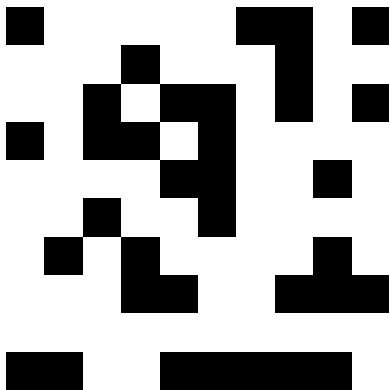
Difícil - 10x10



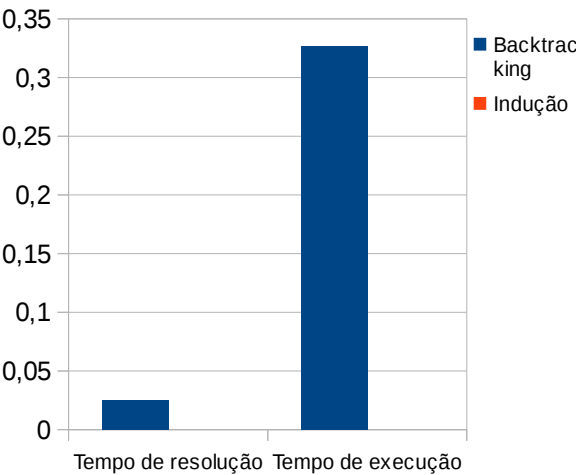
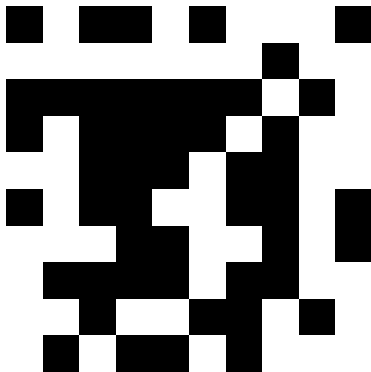
Difícil – 10x10



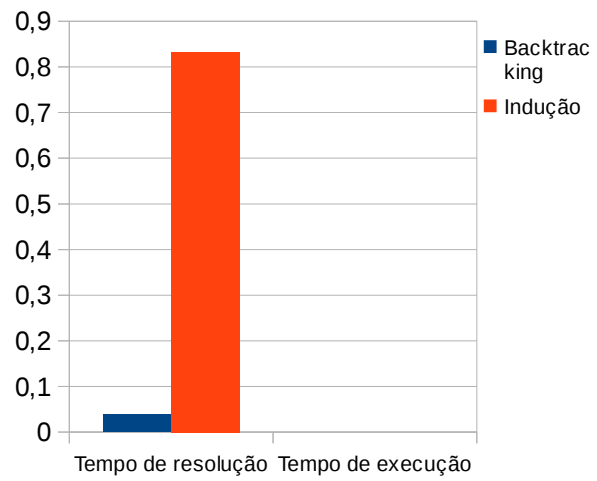
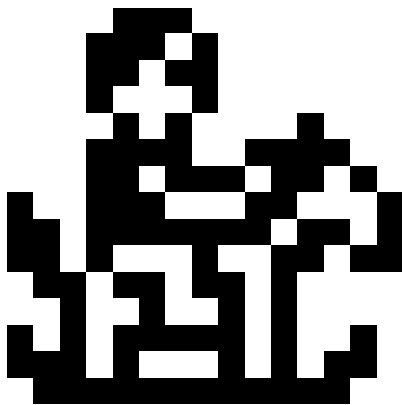
Difícil - 10x10



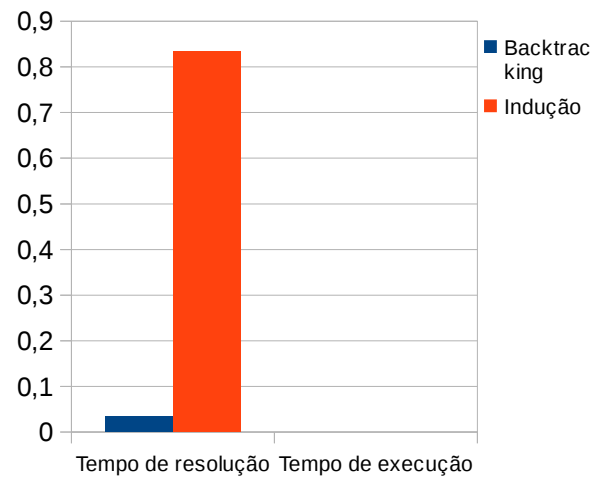
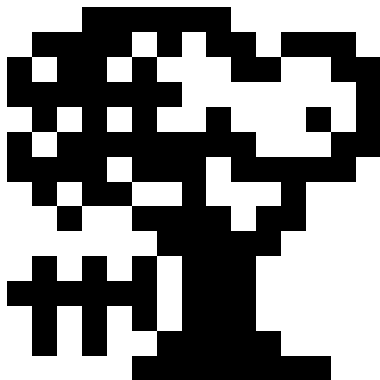
Difícil – 10x10



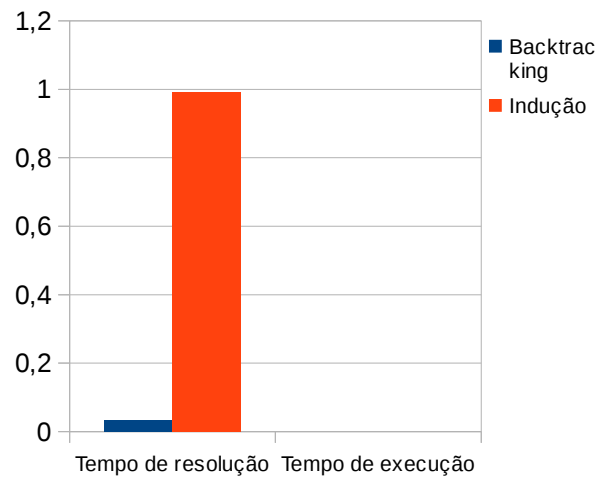
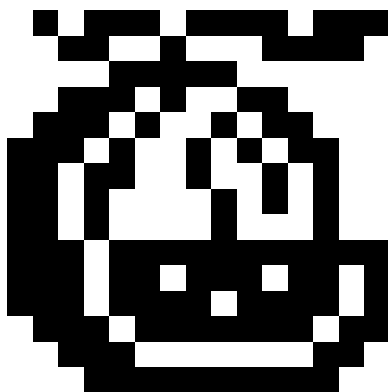
Facíl - 15x15



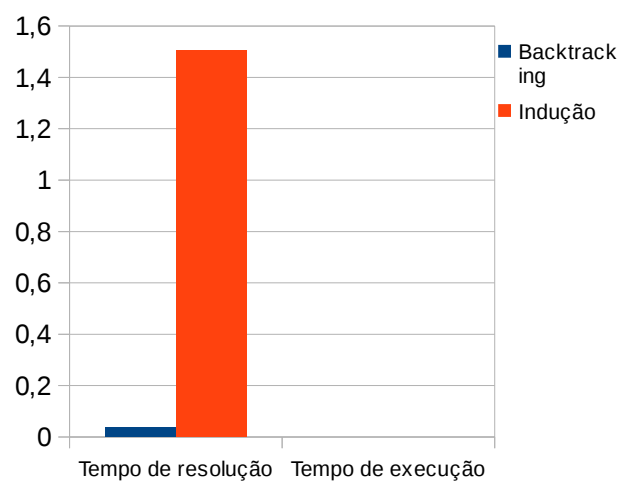
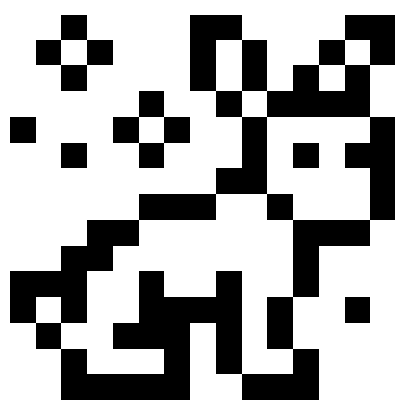
Facíl – 15x15



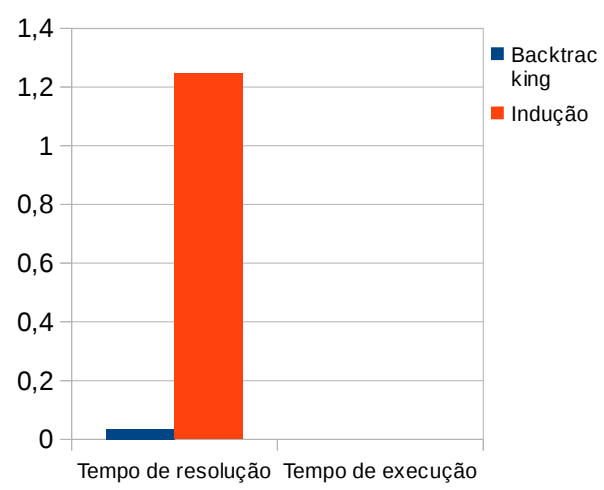
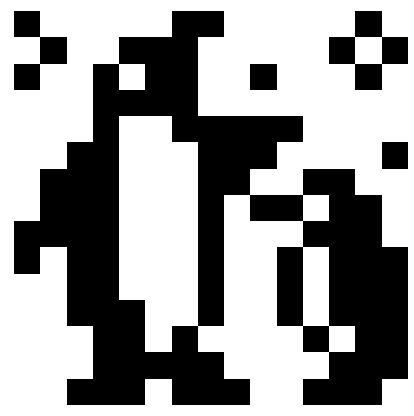
Facíl – 15x 15



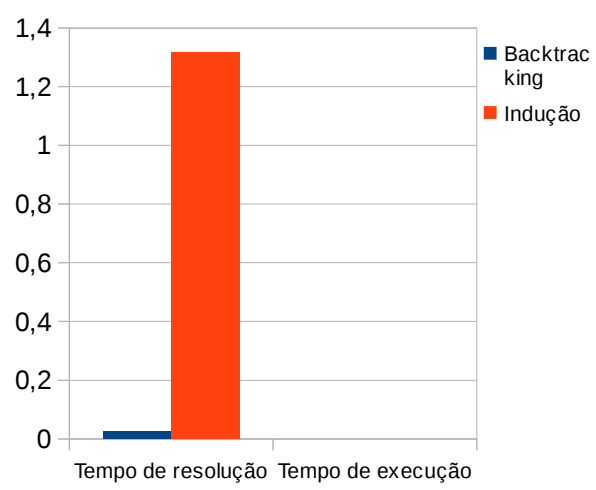
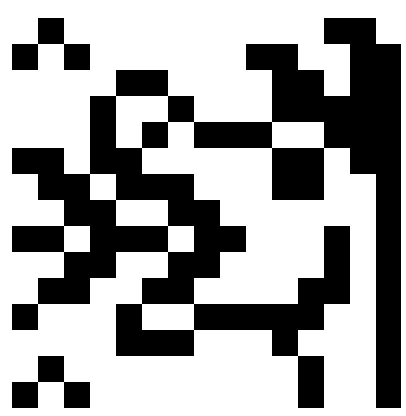
Facíl – 15x15



Facíl – 15x15

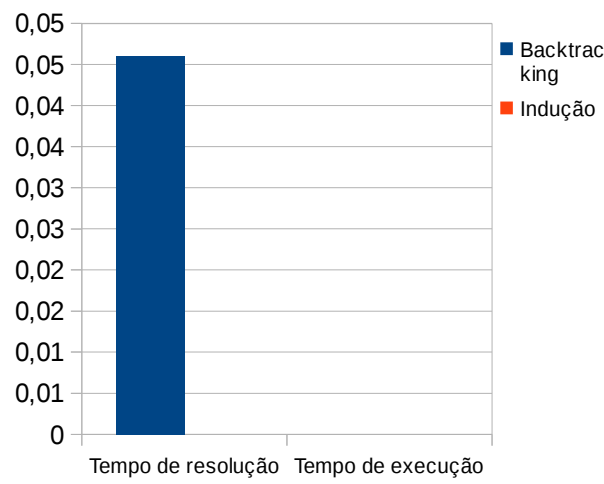
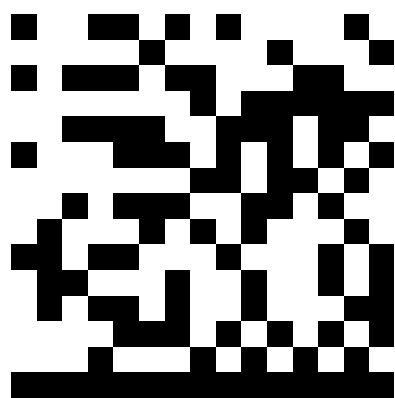


Facíl – 15x15

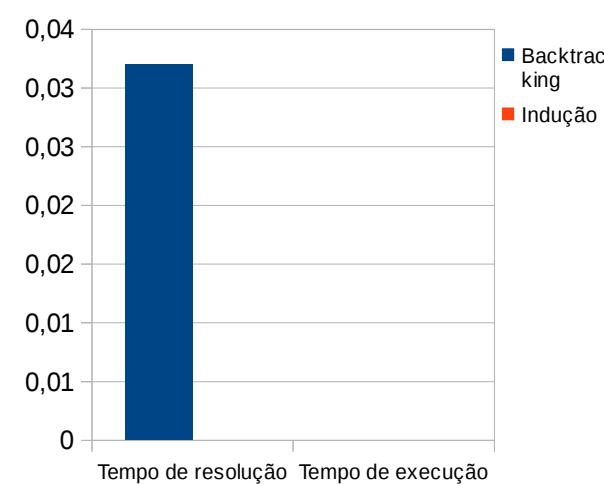
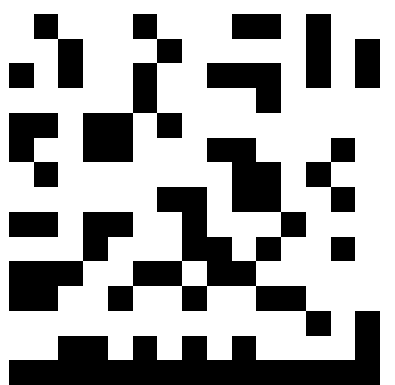




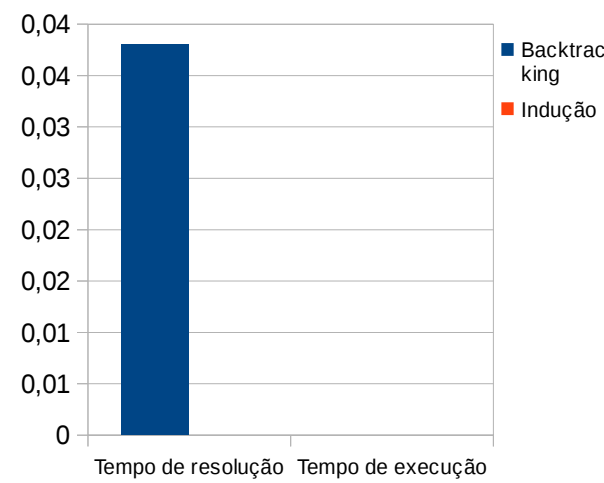
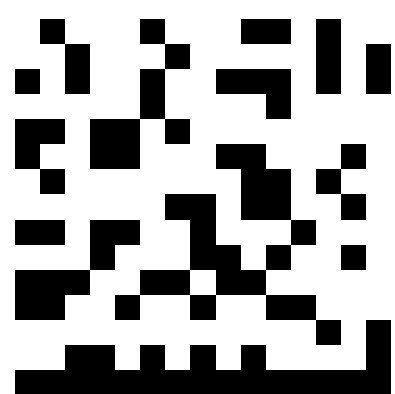
Difícil – 15x15



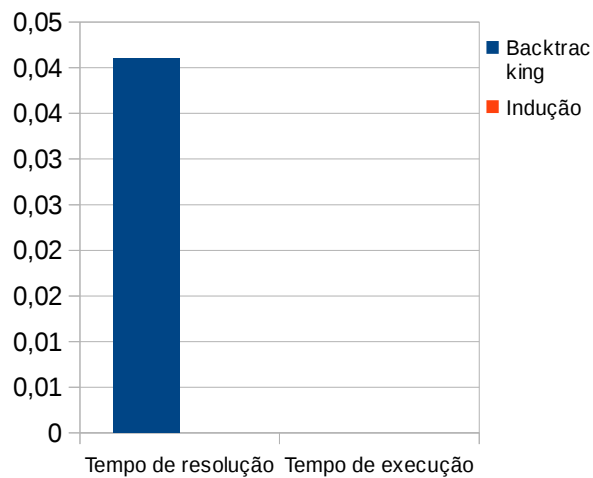
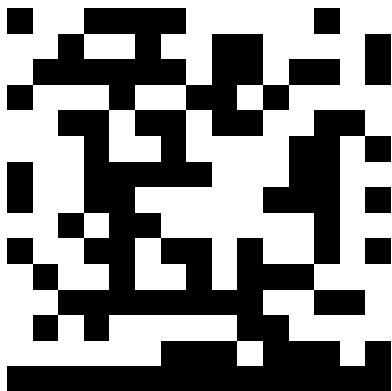
Difícil – 15x15



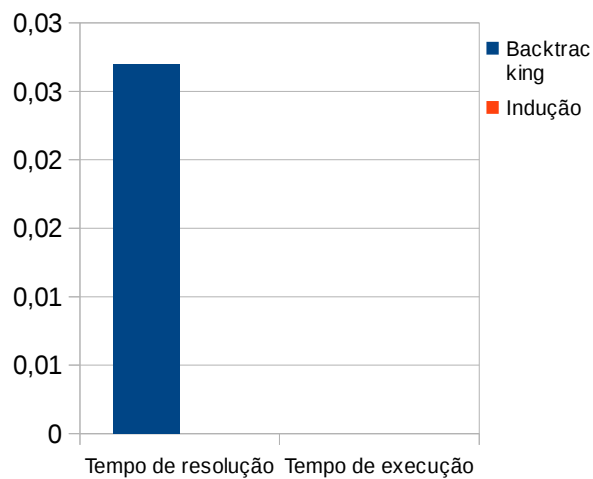
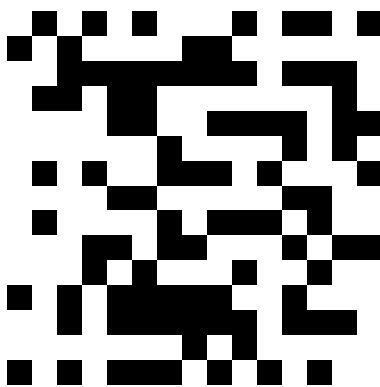
Difícil – 15x15



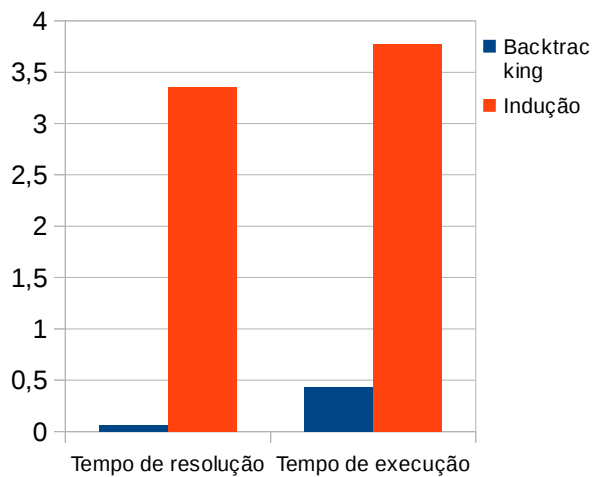
Difícil – 15x15



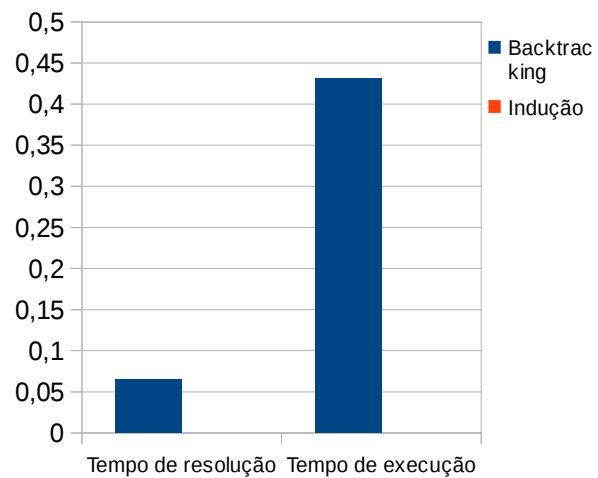
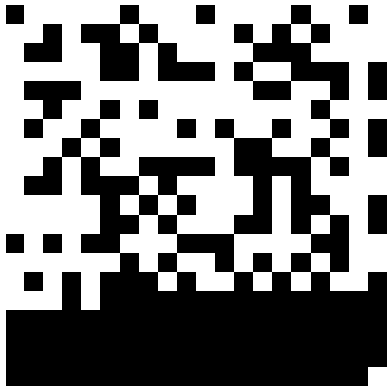
Difícil – 15x15



Facíl – 20x20



Difícil – 20x20



#### 4. Ambiente de Testes

Os testes foram realizados em DELL Inspiron N5010, com 7,6 Gb de RAM e um processador Intel Core i5 CPU M 460 @ 2.53GHz × 4 e uma Intel Ironlake Mobile.

#### 5. Análise dos Testes

Podemos perceber que o método com backtracking se mostra muito mais eficiente que o método de indução, com mais de 10 vezes a velocidade. Ele também se mostra com maior capacidade de soluções complexas. Mesmo com seu maior consumo de memória sua eficácia extremamente superior se mostra mais que útil.