

Algoritmos e Estruturas de Dados

Ano letivo 2024/2025

**Trabalho 1**

Relatório

Turma: P7

Grupo: 114250 - 112726

**114250** – Tiago Alexandre Rodrigues Ramos

**112726** – ­Bruno Miguel Borlido Pereira

# **Introdução**

Este relatório tem como objetivo analisar computacionalmente as funções ImageCreateChessboard(…) e ImageAnd(…) para sua melhor compreensão. Nele irá ser abordado conceitos discutidos em aula como complexidade de algoritmos e analise formal de algoritmos.

# **Análise Formal de ImageCreateChessboard(…)**

A função ImageCreateChessboard(…) cria uma imagem com quadrados alternados a preto e branco como um tabuleiro de xadrez. A seguir na Tabela A estão presentes os dados de vários testes em relação ao espaço que a imagem criada ocupa:

|  |
| --- |
| w = width  h = height  s = square\_edge  Nº de runs = w/s  Size of an Integer = 4 bits  Size = h\*(w/s + 2)\*4 bits |

Assumindo:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | H | S | Size |  | W | H | S | Size |
| 4 | 4 | 1 | 96 | 16 | 8 | 2 | 320 |
| 8 | 4 | 1 | 160 | 4 | 16 | 2 | 256 |
| 16 | 4 | 1 | 288 | 8 | 16 | 2 | 384 |
| 4 | 8 | 1 | 192 | 16 | 16 | 2 | 640 |
| 8 | 8 | 1 | 320 | 4 | 4 | 4 | 48 |
| 16 | 8 | 1 | 576 | 8 | 4 | 4 | 64 |
| 4 | 16 | 1 | 384 | 16 | 4 | 4 | 96 |
| 8 | 16 | 1 | 640 | 4 | 8 | 4 | 96 |
| 16 | 16 | 1 | 1152 | 8 | 8 | 4 | 128 |
| 4 | 4 | 2 | 64 | 16 | 8 | 4 | 192 |
| 8 | 4 | 2 | 96 | 4 | 16 | 4 | 192 |
| 16 | 4 | 2 | 160 | 8 | 16 | 4 | 256 |
| 4 | 8 | 2 | 128 | 16 | 16 | 4 | 384 |
| 8 | 8 | 2 | 192 |

Tabela , Tamanhos em bits de testes com várias imagens

A imagem com mais runs ocorre quando S = 1, o número de runs é W.

A imagem com menos runs ocorre quando S = W, o número de runs é 1.

Gráfico

Gráfico

Gráfico

# **Análise Formal de ImageAND(…)**

A função ImageAND(…) usa duas imagens como input e cria uma terceira imagem, sendo esta o resultado da operação lógica AND de cada um dos pixéis das imagens de input.

|  |
| --- |
| w = width  h = height  n = nº de pixéis da imagem(n=w\*h) |

Assumindo:

O Nº de execuções do código principal do algoritmo é igual a quantidade de pixéis da imagem

Organização dos dados não afeta a complexidade do algoritmo

Ordem de complexidade = Θ(n), linear

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W\H | 4 | 5 | 6 | 7 |
| 4 | 16 | 20 | 24 | 28 |
| 5 | 20 | 25 | 30 | 35 |
| 6 | 24 | 30 | 36 | 42 |
| 7 | 28 | 35 | 42 | 49 |

Tabela

Gráfico

No melhor caso possível o algoritmo é executado 1 vez.

No pior caso possível o algoritmo é executado n vezes.

# **Conclusão**

De forma sucinta, ao desenvolver este relatório, foi-nos possível explorar as funções **ImageCreateChessboard(…)** e **ImageAnd(…)**, analisando o seu funcionamento e a variação de suas complexidades. No caso da função **ImageCreateChessboard(…)**, foi observado como as variáveis **width**, **height** e **square\_edge** influenciam o tamanho final da imagem e o número de execuções (runs), sendo também validada a fórmula para o cálculo do espaço ocupado pela imagem em bits. Para a função **ImageAnd(…)**, verificou-se que o número de execuções do algoritmo é proporcional ao número de píxeis da imagem resultante, apresentando uma complexidade linear Θ(n). Constatou-se ainda que essa complexidade não é afetada pela organização dos dados, mas apenas pelas dimensões das imagens de entrada.

Essa análise auxiliou o entendimento de como os algoritmos funcionam na prática, evidenciando a importância de avaliar a sua eficácia e o impacto computacional.