

# Movimento do Cavalo

Trabalho de Algoritmos e Estrutura de Dados 2

1<sup>st</sup> Caetano Chinarelli Souza *Regitro Acadêmico 2344955*

*Universidade Tecnológica Federal do Paraná*

Pato Branco, Paraná

2<sup>nd</sup> Guilherme Iago Marcante Della Libera *Regitro Acadêmico 2199572*

*Universidade Tecnológica Federal do Paraná*

Pato Branco, Paraná

3<sup>rd</sup> Kelvyn Augusto Waltrick Nonato *Regitro Acadêmico 2345048*

*Universidade Tecnológica Federal do Paraná*

Pato Branco, Paraná

4<sup>th</sup> Luiz Eduardo Rufatto *Regitro Acadêmico 2079933*

*Universidade Tecnológica Federal do Paraná*

Pato Branco, Paraná

5<sup>th</sup> Vinicius Soares do Rosario *Regitro Acadêmico 2247305*

*Universidade Tecnológica Federal do Paraná*

Pato Branco, Paraná

## I. INTRODUÇÃO

Com o tempo, a programação foi se desenvolvendo cada vez mais rápido ao redor do mundo, e com isso foi se criando das mais diversas técnicas para a resolução de muitos problemas existente, e com isso conseguindo cada vez mais a otimização de códigos para seu melhor funcionamento e execução.

No problema que foi escolhido para resolvermos, o deslocamento de um cavalo em um tabuleiro de xadrez, em que é necessário achar a melhor combinação de passos possíveis para chegar até a posição final definida pelo jogador. Com isso, se encaixa a otimização de resultados, fazendo com que o processamento seja cada vez menor, buscando o melhor caso entre todos os possíveis.

Ao fazer a escolha do melhor modelo para resolver o problema selecionado, foi analisado qual seria a ideia para melhor aproveitamento de memória e processamento, onde o backtracking acabou se destacando por ser um modelo que descarta soluções mediana ou péssimas sem mesmo terminar sua checagem, e escolhendo sempre o melhor caso para a resolução do problema.

Após o problema resolvido e a concretização da melhor ação a ser tomada, a possibilidade da resolução de outros tipos de problema desse meio utilizando essa mesma técnica poderá ser feita com muito mais rapidez e economizando memória e processamento da máquina que está sendo utilizada.

## II. PROBLEMÁTICA

### A. Descrição do problema

O problema tratado foi o do deslocamento do cavalo no tabuleiro de xadrez. Esta proposta, segue a seguinte descrição (conforme consta nas especificações do trabalho): dado um tabuleiro de ordem  $N$  ( $N \times N$ ) e uma posição inicial ( $X_0, Y_0$ ),

o algoritmo deve encontrar uma solução ótima (caso exista) com a menor quantidade de movimentos para chegar a posição final. O padrão de movimento do cavalo em um jogo de xadrez normal deve ser seguido, isto é, “em L”. Além disso, deve ser impresso a quantidade de passos e uma matriz onde cada elemento indica o número de passos para chegar a posição em questão.

A escolha deste problema resumiu-se a um acordo comum entre todos os integrantes do grupo através de uma votação, levando em consideração o consenso entre os membros que este foi de fácil compreensão, o que permitiu um planejamento mais claro e eficiente quanto a como abordá-lo.

### B. Motivação para a escolha do problema

Texto aqui

### C. Estratégias para a solução

Do mesmo modo que são tratados diversos problemas envolvendo algoritmos, é possível encontrar múltiplas soluções através de diferentes estratégias aplicadas. O critério de desempate, neste caso e em geral, reduz-se a escolha do algoritmo com melhor eficiência em comparação aos demais, ou seja, que possui o menor custo computacional. Ainda para a escolha, consideram-se a facilidade de manipular a estratégia para a aplicação desejada e a legibilidade do código. A partir disso, foi decidido a escolha de uma estratégia comprovadamente eficiente para a resolução do cenário em questão: backtracking. Backtracking é um tipo de algoritmo que deriva da busca por força bruta, sendo um refinamento desta. A estratégia em questão é utilizada para encontrar todas ou algumas soluções para problemas computacionais, especialmente problemas de satisfação de restrições. Consiste em construir candidatos para

as soluções, “abandonando” um candidato (origina o termo backtrack, traduzindo, “retroceder”) ao determinar que não pode ser completado para uma solução válida. Isto implica que o algoritmo só é válido para problemas que admitem “soluções parciais” como candidatos para soluções, então possibilitando um teste de validez relativamente simples para que a solução possa ser completada. Conceitualmente, backtracking é frequentemente representado por uma estrutura de árvore, seguindo o padrão de busca em profundidade. Os candidatos a solução são os nós da árvore e cada nó de solução parcial é nó pai dos nós candidatos a solução, portanto também são nós intermediários. Se um nó candidato parcial é determinado como inválido, toda a subárvore desse nó também é desconsiderada, então a busca retrocede para o nó anterior e repete o processo de verificação para os demais nós da estrutura. A árvore é percorrida recursivamente, garantindo que toda a árvore válida é percorrida, mas não toda a árvore potencial (que contém todos os possíveis candidatos mas nem todos válidos).

REFERÊNCIAS  
REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

texto aqui

III. DESCRIÇÃO DAS SOLUÇÕES DO PROBLEMA

Texto aqui

IV. ANÁLISE DE COMPLEXIDADE TEMPO E DE ESPAÇO

Texto aqui

```
1  #include <stdio.h>
2  int main() {
3      printf("Ola, Mundo!\n");
4      return 0;
5  }
```

TABLE I  
TABLE TYPE STYLES

Table Head	Table Column Head		
	Table column subhead	Subhead	Subhead
copy	More table copy <sup>a</sup>		

<sup>a</sup>Sample of a Table footnote.



Fig. 1. Example of a figure caption.

V. CONCLUSÃO

Texto aqui

VI. DECLARAÇÃO DE AUTORIA

Relatorio dos membros aqui