

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
PROJETO E ANÁLISE DE ALGORITMOS

1^o semestre de 2025

Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 1

Data de Entrega: 22 de Abril 2025.

Trabalho Dupla

Este trabalho tem por objetivo exercitar primitivas básicas da Linguagem C e iniciar a discussão sobre problemas complexos e a sua solução. Além disso, esse trabalho também visa aplicar alguns paradigmas de programação.

Dama com Vovô

Mesmo com poucos dias de férias, como é tradicional de sua família, você resolveu passar alguns dias com seu avô. Ele já está bem velhinho e sempre fica muito feliz com suas visitas. Uma das diversões preferidas do vovô é jogar dama com você, não apenas jogar, mas te dar uma verdadeira surra. Depois desses poucos dias com seu avô e de várias surras na dama, você resolveu utilizar seus conhecimentos em Ciência da Computação para tentar surpreender seu velho amigo na próxima folga (E você achava que PAA não te serviria para nada né?)

O Jogo de Damas, inventado a mais de 4.000 anos atrás possui várias variações ao redor do mundo. Trata-se de um jogo de tabuleiro para dois jogadores, cujo tabuleiro é quadrado com 8x8 pequenos quadrados que são alternadamente coloridos com uma cor clara e uma cor escura, no padrão usual de um tabuleiro de damas (os quadrados de cor escura é que são as casas). O Jogo de damas jogados entre você e seu avô é um porquinho diferente. O tabuleiro é retangular, com N linhas e M colunas, ou seja, nem sempre é um tabuleiro quadrado e o número de pequenos quadrados pode mudar a cada novo jogo (seu avô gosta de inovar e surpreender para tornar mais divertida a brincadeira!).

No início do jogo, cada jogador (você e seu avô) tem um certo número de peças, posicionadas nas casas mais próximas da borda do tabuleiro que cada um escolher (sempre bordas opostas). Durante o jogo, as peças só podem ocupar as casas do tabuleiro. Um dos movimentos do jogo é capturar uma peça do oponente (vovô chama de comer a peça), saltando sobre ela, diagonalmente, para a casa adjacente além da peça, casa esta que deve estar vazia. A peça do oponente é então removida do tabuleiro. As três casas envolvidas na captura (a casa inicial, a que contém a peça do oponente e a vazia, onde sua peça estará após a jogada) devem estar diagonalmente alinhadas e devem ser diagonalmente adjacentes. Na Dama do Vovô uma peça pode capturar peças do oponente saltando diagonalmente para a frente ou para trás (seu avô é o cara!). Você pode também efetuar uma captura múltipla, com uma peça apenas, saltando seguidamente para casas vazias sobre peças oponentes, podendo mudar de direção, saltando primeiro em uma direção e depois em outra. Você pode capturar apenas uma peça a cada salto, mas pode capturar várias peças com saltos seguidos. Você não pode saltar sobre uma peça sua, e não pode saltar a mesma peça oponente mais de uma vez.

Dadas as dimensões do tabuleiro e uma descrição do estado corrente de um jogo, na sua jogada você deve determinar computacionalmente o número máximo de peças do seu oponente que podem ser capturadas em um movimento de captura. E claro, você não pode demorar muito a determinar sua jogada!

Assim, seu objetivo nesse trabalho é implementar duas estratégias para solucionar esse problema:

- Uma estratégia força bruta;
- Um estratégia alternativa que seja capaz de sempre resolver o problema, tentando o mínimo de movimentos;

Entrada e Saída

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois inteiros N e M indicando respectivamente o número de linhas e o número de colunas do tabuleiro ($3 \leq N \leq 20$, $3 \leq M \leq 20$ e $N \times M \leq 200$). O quadrado mais à esquerda do tabuleiro na borda mais próxima ao jogador é uma casa. A segunda linha contém a descrição do estado do jogo.

Cada descrição consiste de $(N \times M)/2$ inteiros, separados por um espaço, correspondendo às casas do tabuleiro, que são numeradas de 1 a $(N \times M)/2$, da esquerda para a direita, da borda mais próxima ao jogador à borda mais próxima ao seu oponente. Na descrição do estado do jogo, 0 representa uma casa vazia, 1 representa uma casa com uma de suas peças, e 2 representa uma casa com uma peça de seu oponente. Há no máximo $(N \times M)/4$ peças de cada jogador no tabuleiro. O final da entrada é indicado por $N = M = 0$.

Exemplo de entrada:

```
3 3
2 1 2 0 1
5 3
1 0 2 1 0 2 0 0
8 8
2 2 2 2 0 0 0 0 2 2 2 2 0 0 0 0 2 2 2 2 0 0 0 0 2 2 2 2 0 1 0 0
0 0
```

Exemplo de saída

```
1
2
7
```

O arquivo executável deve ser chamado de `tp1` e deve receber como parâmetro apenas o arquivo de entrada:

```
./tp1 -i entrada.txt
```

Na tela, o programa deve imprimir apenas os tempos de usuário e os tempos de sistema para comparação. Para avaliação do tempo, utilize as funções *getrusage* e *gettimeofday*.

Documentação

Deve ser clara e objetiva, descrevendo as soluções adotadas e justificando bem as escolhas realizadas. Devem possuir também uma análise de complexidade detalhada das soluções. Em termos de análise de resultados, avalie o desempenho e funcionamento de seus algoritmos para

diversas configurações e avalie também o tempo de execução dos mesmos (compare-os). Lembre-se, o importante é você apresentar maturidade técnica em suas discussões.

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especificação caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp1** e deve receber como parâmetro apenas o nome do arquivo de entrada de dados. Não serão aceitos outros nomes de executáveis além dos mencionados.
- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo::
 - descrição das soluções e estruturas de dados utilizadas;
 - análise da complexidade das rotinas;
 - análise dos resultados obtidos.
 - a documentação não pode exceder 12 páginas.

Distribuição dos pontos:

- execução (E)
execução correta: 80%
- estilo de programação
código bem estruturado: 10%
código legível: 10%

- documentação (D)
 - comentários explicativos: 40%
 - análise de complexidade: 30%
 - análise de resultados: 30%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$