

Trabalho Prático 1

Algoritmos I

Entrega: 01/09/2020

1 Introdução

Adriana e suas amigas são apaixonadas por jogos de tabuleiro. Em um dia qualquer, Adriana descobre um jogo muito interessante: O **Jogo do Pulo**. Ansiosa por compartilhar o jogo com suas amigas, imediatamente marca um encontro para jogarem. Já inicia o encontro explicando o jogo. O objetivo do Jogo do Pulo é bem simples: dado um tabuleiro, jogadores e suas posições iniciais, atingir a última casa do tabuleiro (em um tabuleiro de tamanho $N \times M$, essa seria a casa $N-1 \times M-1$).

O tabuleiro pode conter qualquer número de dimensões, dependendo do modo do jogo e da realidade em que está inserido. Neste trabalho iremos abordar o modo bi-dimensional do jogo. Cada célula do tabuleiro contém um número que informa quantas casas o jogador deve pular partindo dela. Note que é o número exato de casas, e não o máximo partindo dela. O tabuleiro possui números inteiros positivos, e pode conter casas com o número zero (no caso do número da casa ser um zero, o jogador nunca consegue sair dela).

Os personagens movimentam-se apenas em sentidos paralelos aos limites do tabuleiro. No caso de um tabuleiro bidimensional, por exemplo, o personagem só pode andar nas direções x ou y, ou seja, movimentos diagonais não são permitidas. Além disso, o movimento pode se dar em qualquer sentido, desde que seguindo a restrição anterior. Novamente, no caso bi-dimensional, o personagem pode se movimentar nas direções +x, -x, +y ou -y.

A ordem de jogadas e a casa inicial dos jogadores é definida no começo do jogo. No nosso caso, essa ordem e as casas iniciais serão dadas como entrada do problema.

Definidas as posições iniciais, vamos começar o jogo. Seguindo a ordem posteriormente sorteada, cada jogador pula o número de casas escrito na sua posição, em qualquer sentido permitido e desejado. A ordem de jogadas nas rodadas seguintes é definida pelo tamanho do pulo anterior. A regra é que tem maior prioridade aquele que pulou a menor quantidade de casas na rodada anterior e, em caso de empate, tem prioridade o jogador que jogou antes na primeira rodada.

O jogo pode terminar de duas maneiras: i) a vitória de algum jogador; e ii) a impossibilidade de vitória. No primeiro caso, o jogo termina quando o primeiro jogador atinge a última casa do tabuleiro. Já no segundo caso, o jogo acaba quando fica definido que nenhum jogador conseguirá vencer nunca o jogo (jogo sem vencedores).

Um jogo sem vencedores pode ocorrer quando nenhum jogador consegue chegar na última posição do tabuleiro. Isso pode acontecer ao cair em uma casa marcada com o número 0 (e ele não conseguiria sair de lá), ou ao ficar rodando sempre pelas mesmas casas, sem conseguir sair delas.

2 Exemplos de jogo do pulo

Nessa seção iremos provar, passo a passo, três exemplos do jogo do pulo. Note que ao falarmos de posição, nos referimos ao índice da casa em que o jogador se encontra, e não ao valor dessa casa. Fique atento que uma casa na posição 2, pode ter um valor de pulo diferente de 2.

2.1 Exemplo 1

Por questões didáticas vamos começar exemplificando com o tabuleiro 1-dimensional abaixo. No jogo abaixo temos 3 jogadores: A, B e C. A ordem inicial é A, B e C, nas posições iniciais 0, 1 e 2, respectivamente. Os valores associados a cada posição do tabuleiro bem como as posições iniciais dos jogadores e a ordem de jogada são ilustradas abaixo:

Jogo 1 - Posição Inicial

Posição	0	1	2	3	4	5	6	7	8
Tabuleiro	4	1	3	2	2	1	2	1	0
Personagens	A	B	C						
Ordem	1	2	3						

Ao final da primeira rodada, temos os jogadores A na posição 4, B na posição 2 e C na posição 5. Temos, também, que a nova ordem de prioridade de jogadas é definida por B, C e A, visto que B foi o jogador que deu o menor pulo, enquanto A foi o jogador a pular mais casas. Veja abaixo a representação do tabuleiro ao final da primeira rodada.

Jogo 1 - Rodada 1

Posição	0	1	2	3	4	5	6	7	8
Tabuleiro	4	1	3	2	2	1	2	1	0
Personagens			B		A	C			
Ordem			1		3	2			

Dessa forma, o jogador B faz a primeira jogada, andando 3 casas e alcançando assim a posição 5. Seguindo, o jogador C chega à posição 6, andando 1 casa. Por fim, o jogador A desloca-se 2 casas de modo a alcançar, assim como o jogador C, a posição 6. Define-se assim a ordem prioridade de jogadas por C, A e B, respectivamente.

Jogo 1 - Rodada 2

Posição	0	1	2	3	4	5	6	7	8
Tabuleiro	4	1	3	2	2	1	2	1	0
Personagens						B	C, A		
Ordem						3	1, 2		

Finalmente, na terceira rodada, o jogador C (que detém a prioridade de jogadas) atinge o final do tabuleiro, andando somente uma casa. Assim sendo, o jogador C é consagrado vencedor do jogo do pulo na terceira rodada. Note que os jogadores A e B não jogam nessa rodada. Veja abaixo a representação final do tabuleiro.

Jogo 1 - Rodada 3

Posição	0	1	2	3	4	5	6	7	8
Tabuleiro	4	1	3	2	2	1	2	1	0
Personagens						B	A		C
Winner/Looser						L	L		W

2.2 Exemplo 2

Da mesma forma que no jogo anterior, abaixo temos 3 jogadores: A, B e C. Sorteamos inicialmente na mesma ordem disposta anteriormente. Novamente, os jogadores A, B e C escolhem as posições 0, 1 e 2, respectivamente. Abaixo temos a representação inicial do tabuleiro.

Jogo 2 - Posição Inicial

Posição	0	1	2	3	4	5
Tabuleiro	4	2	1	0	4	0
Personagens	A	B	C			

O jogador A inicia o jogo movimentando-se 4 casas. O jogador B move-se 2 casas no mesmo sentido. Por fim, o jogador C movimenta-se 1 casa. Note que os jogadores B e C alcançam a mesma posição (casa 3). E o jogador A chega na casa 4. Como podemos perceber, nenhum jogador sairá vencedor deste jogo. B e C chegaram a uma casa de movimentação zero (0) e não conseguirão nunca sair dela. Já para o jogador A, o único movimento possível é voltar para a casa 0, e não conseguirá nunca sair dessas duas casas.

Jogo 2 - Rodada 1

Posição	0	1	2	3	4	5
Tabuleiro	4	2	1	0	4	0
Personagens				B, C	A	

Jogo 2 - Rodada 2

Posição	0	1	2	3	4	5
Tabuleiro	4	2	1	0	4	0
Personagens	A			B, C		
Winner/Looser	L			L, L		

Como mostrado no tabuleiro acima, o jogo previamente disposto termina sem vencedores.

2.3 Exemplo 3

Por fim, iremos ilustrar abaixo um exemplo bi-dimensional. Neste exemplo temos apenas 2 jogadores: A e B. A ordem de prioridade inicial é definida pela mesma ordem. Sendo assim, o jogador A escolhe como posição inicial (0, 3) e B (1, 2). Abaixo temos a ilustração de como B pode vencer na quarta rodada:

Jogo 3 - Posição Inicial

1	2	2	3 (A)
3	1	1 (B)	2
1	2	2	0

Rodada 1

1 (A)	2	2	3
3	1	1	2
1	2	2 (B)	0

Rodada 2

1	2 (A)	2	3
3	1	1	2
1 (B)	2	2	0

Rodada 3

1	2	2	3
3	1	1	2
1	2 (B, A)	2	0

Rodada 4

1	2	2	3
3	1	1	2
1	2 (A - L)	2	0 (B - W)

DICA: Note que, em todos os exemplos, os jogadores são profissionais e sempre têm sua movimentação de modo a maximizar sua chance de vitória.

3 Objetivo

O objetivo deste trabalho é definir, para um determinado tabuleiro bidimensional, um determinado número de jogadores (com nomes), determinada posição e ordem inicial, qual jogador será o vencedor e em qual rodada.

Podemos assumir que:

- Após escolhidas as posições iniciais, os jogadores adotarão sempre a melhor estratégia (aquela que lhe dará maiores chances de vencer);
- Como se trata de um jogo bidimensional, os jogadores podem pular nas direções $+x$, $-x$, $+y$ ou $-y$;
- O pulo deve sempre respeitar os limites do tabuleiro (claro que um jogador não pode pular para fora do tabuleiro);
- Para um tabuleiro de dimensões $M \times N$, a última casa é a casa $(M - 1, N - 1)$;
- A ordem na qual os jogadores são apresentados, é a ordem inicial (primeiro jogador apresentado é o primeiro a jogar).

Importante ficar atento:

- Cuidado com loops na solução (jogador que fica pulando entre as mesmas duas casas, por exemplo);
- Evite calcular caminhos de jogadores que não tem chances de ganhar (se um jogador A chega na casa X em alguma rodada, um outro jogador que atinja essa mesma casa em rodadas posteriores tem alguma chance de ganhar o jogo?);
- Pense que existe uma prioridade a ser seguida ao decidir quem será o próximo a pular, e qual a melhor estrutura de dados a ser usada para isso;
- Pense em que situações o programa deve assumir que não existem vencedores;
- Pesquise sobre BFS em grafos, e veja as possíveis modificações para este problema específico.

4 Os arquivos de entrada

Exemplo das linhas da entrada

```
N M // dimensão do tabuleiro;
K // quantidade de jogadores
X11 X12 .. X1M //tabuleiro
X21 X22 .. X2M
.
.
.
XN1 XN2 .. XNM
Ax Ay // posição inicial do jogador A
Bx By
.
.
.
Kx Ky // posição inicial do jogador K
```

5 Os arquivos de saída

Exemplo das linhas da saída - Jogo com vencedor

```
X // Jogador vencedor
R // Número da rodada final
```

Exemplo das linhas da saída - Jogo sem vencedores

```
SEM VENCEDORES
```

6 Exemplo Prático de Entrada e Saída

Exemplo prático da entrada - JOGO 1

```
1 9
3
4 1 3 2 2 1 2 1 0
0 0
0 1
0 2
```

Exemplo prático da saída - JOGO 1

C
3

Exemplo prático da entrada - JOGO 2

1 6
3
3 2 1 0 1 0
0 0
0 1
0 2

Exemplo prático da saída - JOGO 2

SEM VENCEDORES

Exemplo prático da entrada - JOGO 3

4 3
2
1 2 2 3
3 1 1 2
1 2 2 0
0 3
1 2

Exemplo prático da saída - JOGO 3

B
4

7 Avaliação Experimental

Para a avaliação experimental o aluno deverá criar entradas para o problema de forma a avaliar se o tempo de execução do algoritmo está de acordo com a complexidade reportada na documentação, aumentando o tamanho do tabuleiro, guardando os tempos de execução e colocando esses valores em gráficos para melhor visualização.

Podemos assumir um número pequeno de jogadores (11). Não se preocupe em variar isso na análise de tempo.

É necessário executar mais de uma vez o algoritmo para entradas com o mesmo N e M, guardando os tempos de execução e reportando a média e o desvio padrão destes tempos. Faça isso para vários valores diferentes de N e M, para visualizar bem como o tempo está variando com a variação do tamanho da entrada.

Além disso, explique o que acontece com o tempo em jogos sem vencedores.

8 O que deve ser entregue

Você deve submeter um arquivo compacto (zip ou tar.gz) no formato **seu_nome_sua_matrícula** via Moodle contendo:

- todos os arquivos do código *.c*, *.cpp* e *.h* que foram implementados,
- um arquivo *makefile*¹ **que crie um executável `tp1`**,
 - **ATENÇÃO:** O makefile é para garantir que o código está sendo compilado corretamente, de acordo com o modo que vocês modelaram o programa. É **essencial** que ao digitar “make” na linha de comando dentro da pasta onde reside o arquivo makefile, o mesmo compile o programa e gere um executável chamado **`tp1`**.
- sua documentação.

Sua documentação deve ter até 10 páginas contendo:

- uma breve descrição do problema,
- explicações das estruturas de dados e dos algoritmos utilizados para resolver o problema. Para tal, artifícios como pseudo-códigos, exemplos ou diagramas podem ser úteis. Note que documentar uma solução não é o mesmo que documentar seu código. Não inclua textos de códigos em sua documentação.
- análise de complexidade da solução proposta (espaço e tempo). Cada complexidade apresentada deverá ser devidamente justificada para que seja aceita.
- avaliação experimental.

O seu TP deverá ser entregue de acordo com a data especificada no moodle. A penalidade em porcentagem para os TPs atrasados é dada pela fórmula $2^{d-1}/0.16$.

9 Implementação

9.1 Linguagem, Ambiente e Parâmetros

O seu programa deverá ser implementado na linguagem **C** ou **C++** e poderá fazer uso de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de bibliotecas que não a padrão não serão aceitos. Usaremos um ambiente Linux para rodar os códigos.

¹https://pt.wikibooks.org/wiki/Programar_em_C/Makefiles

ATENÇÃO: O arquivo da entrada deve ser passado como parâmetros para o programa através da linha de comando (e.g., `$./tp1 jogo_do_pulo.txt`) e imprimir a saída no **stdout** (com `printf/cout`), não em um arquivo.

ATENÇÃO: certifique-se que o programa execute com os três arquivos passados como exemplos junto com a documentação no Moodle (dataset.zip), sem alterar os arquivos passados. Isso vai garantir que seu programa vai conseguir ler corretamente os arquivos do corretor automático.

9.2 Testes

A sua implementação passará por um processo de correção automática, o formato da saída de seu programa deverá ser idêntico aquele descrito nas seções anteriores. Saídas diferentes serão consideradas erro para o programa. Para auxiliar na depuração do seu trabalho, será fornecido um pequeno conjunto de entradas e suas saídas. É seu dever certificar que seu programa atenda corretamente para qualquer entrada válida.

9.3 Qualidade do código

É importante prestar atenção para a qualidade do código, mantendo-o organizado e comentado para não surgir dúvidas na hora da correção. Qualquer decisão que não estiver clara dada a documentação e a organização do código será descontada na nota final.

10 Consideração Final

Assim como em todos os trabalhos dessa disciplina é estritamente proibido a cópia parcial ou integra de códigos, seja da internet ou de colegas. Seja honesto! Você não aprende nada copiando código de terceiros. Se a cópia for detectada, sua nota será zerada e o professor será informado para que as devidas providências sejam tomadas.