

## VI BXComp

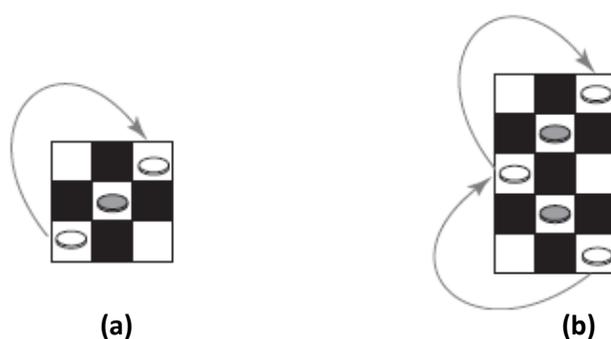
6º Campeonato de Programação para Calouros do Curso de Sistemas de Informação 2016

### 7ª Etapa – Desafio 1

#### O Jogo da Captura

Alan Dulling é um pesquisador de destaque na área de Inteligência Artificial (IA), conhecido por desenvolver sistemas que aplicam IA em jogos, como dominó, palavras-cruzadas e forca, os quais são capazes de vencer qualquer oponente humano, até mesmo os campeões mundiais nessas modalidades. Recentemente, ele foi desafiado pelo atual campeão mundial em uma versão específica do jogo de damas, Eric Enigman, invicto nos últimos 10 anos, a desenvolver um sistema que seja capaz de vencê-lo.

A versão do jogo de damas que o sistema deve simular possui as seguintes especificações: i) o tabuleiro possui  $L$  linhas e  $C$  colunas - totalizando  $L \times C$  casas; ii) as casas do tabuleiro são alternadamente coloridas, com uma cor clara e uma cor escura, iniciando-se por uma casa de cor clara na extremidade inferior esquerda do tabuleiro – coordenadas  $(L-1, 0)$ , ou seja, a coloração é feita da esquerda para a direita (colunas), de baixo para cima (linhas); iii) as peças só podem estar posicionadas em casas de cor clara; iv) as peças podem se mover diagonalmente para frente ou para trás; v) para realizar a captura de uma peça, as três casas envolvidas – a casa que contém sua peça, a casa que contém a peça do oponente e a casa vazia – devem estar diagonalmente alinhadas e serem diagonalmente adjacentes; vi) a cada salto, somente uma peça pode ser capturada, isto é, saltos sobre mais de uma peça do oponente não são permitidos; vii) é possível capturar várias peças com saltos seguidos, em uma captura múltipla. Nesta, a sua peça pode mudar de direção; viii) quando uma peça é capturada, ela é retirada do tabuleiro; e ix) você não pode saltar sobre uma peça sua.



**Figura 1** – Captura de uma peça em um tabuleiro de 3x3 (a); e captura múltipla de 2 peças em um tabuleiro 5x3 (b).

Dulling está na etapa final do desenvolvimento do sistema que será capaz de vencer Enigman, necessitando apenas finalizar a implementação do algoritmo que maximiza a captura de peças em uma jogada considerando-se capturas múltiplas e, para isso, solicitou a sua ajuda.

## Tarefa

Dadas as dimensões do tabuleiro e uma descrição do estado corrente de um jogo, você deve implementar um programa que determine o número máximo de peças do seu oponente que podem ser capturadas em uma jogada múltipla, assim como as coordenadas das casas que a sua peça deve visitar para capturá-las. Caso mais de uma jogada resulte em um número máximo igual de peças capturadas, a jogada de maior impacto deverá ser a escolhida.

Definimos o impacto de uma jogada como:  $\sum_{i=1}^{Q+1} i * (x_i + y_i)$ , em que  $i$  representa cada casa pela qual sua peça vai passar nas jogadas,  $Q$  a quantidade máxima de saltos em uma captura múltipla e  $(x_i + y_i)$  o valor da soma para as coordenadas  $x$  e  $y$  em que a sua peça estará localizada na casa  $i$  do tabuleiro, representando linha e coluna, respectivamente. Por exemplo, na Figura 2, há apenas um salto possível em ambas as possibilidades de jogada (logo,  $Q = 1$ ). Sua peça, porém, passará por duas casas: a inicial e a final após o movimento de captura. Assim, a jogada escolhida seria aquela em que a peça iria para a diagonal inferior direita, pois  $(1 * (2 + 0)) + (2 * (0 + 2)) < (1 * (2 + 0)) + (2 * (4 + 2))$ . No cálculo realizado para a jogada de maior impacto,  $(1 * (2 + 0))$  representa a peça antes do primeiro movimento de captura ( $i = 1$ ), localizada nas coordenadas (2,0), que é somado a  $(2 * (4 + 2))$ , representando a peça após o primeiro movimento de captura ( $i = 2$ ), localizada nas coordenadas (4,2).

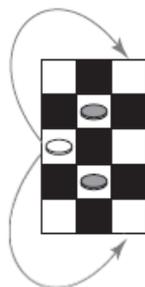


Figura 2 – Jogadas para determinado estado que resultam em valor máximo igual de peças capturadas

## Entrada

A entrada será composta por vários casos de testes. Cada um deles consiste de duas linhas: a primeira linha contém dois inteiros  $L$  e  $C$  indicando respectivamente o número de linhas e o número de colunas do tabuleiro, sendo  $3 \leq L, C \leq 20$ ; a segunda linha contém a descrição do estado do jogo. Cada descrição consiste de  $\lceil (L * C) / 2 \rceil$  inteiros, separados por um espaço simples, representando as peças que ocupam as casas do tabuleiro, e são listadas da esquerda para a direita, de baixo para cima, conforme a Figura 3. Na descrição do estado do jogo, poderão constar apenas três números inteiros: o número 0, que representa uma casa vazia; o 1, que representa que a casa está ocupada por uma de suas peças; e o 2, que representa que a casa está ocupada por uma peça de seu adversário. Há, no máximo,  $\lfloor (L * C) / 4 \rfloor$  peças de cada jogador. O final da entrada é indicado por  $L = C = 0$ .

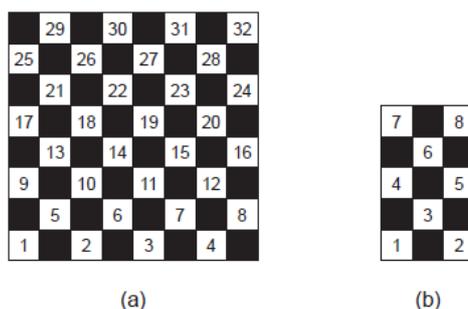


Figura 3: Ordem na qual as peças são listadas em (a) tabuleiro 8 x 8 e em (b) tabuleiro 5 x 3.

## Saída

Para cada caso de teste, seu programa deve produzir: i) uma linha, contendo um inteiro **N** indicando o maior número de peças de seu oponente que podem ser capturadas em uma jogada; ii) **N+1** linhas, caso **N > 0**, cada qual contendo um par de inteiros **x** e **y**, separados por um espaço simples, representando as coordenadas das casas pelas quais sua peça passou durante a trajetória realizada, em termos de linhas e colunas do tabuleiro, respectivamente, sendo  $0 \leq x < L$  e  $0 \leq y < C$  – a coordenada (0,0) se refere à casa na extremidade superior esquerda e a coordenada (L-1,C-1) se refere à casa na extremidade inferior direita. A primeira linha em (ii) representa a coordenada inicial da trajetória e as demais linhas as casas que foram posteriormente visitadas. Se **N = 0**, não é necessário produzir linhas adicionais, visto que a peça não se movimentou. Após o último caso de teste deverá haver uma quebra de linha.

## Exemplo de Entrada

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

## Exemplo de Saída

```
1
2 0
0 2
9
7 0
5 2
3 0
1 2
3 4
5 2
7 4
5 6
3 4
1 6
0
```