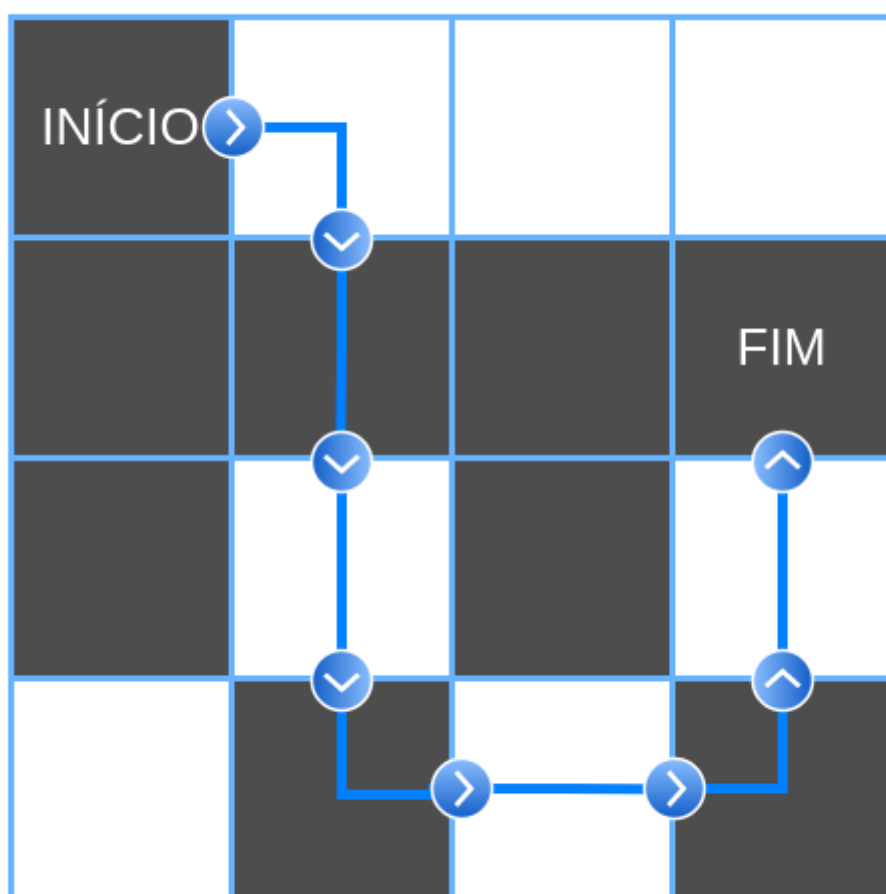


## 🔗 Caminho Binário Alternado

Jogos de tabuleiro, onde o objetivo é encontrar um caminho desde uma posição inicial até uma posição final, são uma excelente opção de entretenimento. Além disso, são ótimos para estimular o raciocínio lógico nas crianças.

O jogo do Caminho Binário Alternado é jogado em um tabuleiro com **L** linhas e **C** colunas, de forma que cada posição do tabuleiro possui uma cor preta ou branca. O objetivo consiste em encontrar um caminho a partir de uma posição inicial até uma posição final, com ambas as posições sendo definidas no início da partida. Entretanto, as posições do caminho devem ter uma cor alternada, ou seja, um caminho válido para o jogo não possui duas posições consecutivas com a mesma cor. Além disso, a movimentação no tabuleiro só pode ser realizada para posições adjacentes nas direções norte, sul, leste e oeste. A imagem a seguir mostra um exemplo de uma partida do jogo do Caminho Binário Alternado e um possível caminho da posição inicial até a posição final.



Note que no exemplo existem múltiplos caminhos válidos. Contudo, dependendo da configuração do tabuleiro e das posições inicial e final escolhidas para a partida, pode ser que não exista nenhum caminho válido. A primeira pessoa que encontrar um caminho binário alternado da posição inicial até a posição final é a vencedora da partida. Durante a partida, qualquer jogador pode declarar que não existe nenhum caminho válido no tabuleiro. Caso a declaração seja verdadeira, a pessoa que fez a declaração é a vencedora. Caso contrário, a pessoa é eliminada da partida.

Como você está desenvolvendo suas habilidades com programação, seus amigos acham que seria uma boa ideia desenvolver um programa para determinar se existe ou não um caminho válido dada uma partida do jogo do Caminho Binário Alternado.

O seu programa receberá como entrada um número inteiro `L` indicando a quantidade de linhas do tabuleiro. Em seguida, seu programa receberá `L` linhas. Cada linha é formada por `C` caracteres `0` ou `1` separados pelo caractere de espaço, onde `0` e `1` representam, respectivamente, uma posição na cor preta e branca. Você pode assumir que cada linha possui a mesma quantidade de caracteres. Por fim, seu programa receberá, na mesma linha, quatro números inteiros `l1`, `c1`, `l2`, `c2`, separados pelo caractere de espaço. A dupla `l1` e `c1` representa a linha e coluna da posição inicial e a dupla `l2` e `c2` representa a linha e coluna da posição final. Neste jogo, o canto superior esquerdo do tabuleiro é representado pela posição `(0, 0)`.

Como saída, o seu programa deverá imprimir uma mensagem indicando o estado final da partida. Caso seja possível encontrar um caminho binário alternado da posição inicial até a posição final, então a seguinte mensagem deve ser exibida:

```
caminho encontrado
```

Caso contrário, a seguinte mensagem deve ser impressa:

```
caminho nao encontrado
```

Exemplos de entradas e saídas esperadas pelo seu programa:

## Teste 01

### Entrada

```
4
0 1 1 1
0 0 0 0
0 1 0 1
1 0 1 0
0 0 1 3
```

Saída

caminho encontrado

## Teste 02

Entrada

```
4
0 1 1 1
0 0 0 0
0 1 0 1
1 0 1 0
0 0 1 0
```

Saída

caminho nao encontrado

## Teste 03

Entrada

```
9
1 1 1 1 1 1 1
1 0 1 0 1 0 1
1 1 1 1 1 1 1
1 0 1 0 1 0 1
1 1 1 1 1 1 1
1 0 1 0 1 0 1
1 1 1 1 1 1 1
1 0 1 0 1 0 1
1 1 1 1 1 1 1
1 1 7 5
```

Saída

## Código Base

---

No arquivo auxiliar lab14.py você irá encontrar um código base para dar início ao processo de elaboração deste laboratório. Para facilitar a implementação do seu programa, no código base existe o cabeçalho de uma função com a descrição do que deve ser desenvolvido. A ideia é que a função desenvolvida para obter uma solução para o problema seja recursiva. Além disso, no código base, também é fornecida a formatação para imprimir a solução do problema.

```
def caminho(tabuleiro, linha_atual, coluna_atual, linha_fim, coluna_fim,
# ...

# ...
print("caminho encontrado")
# ...
print("caminho nao encontrado")
```

## Orientações

---

- Veja [aqui](#) a página de submissão da tarefa.
- O arquivo a ser submetido deve se chamar lab14.py.
- No link "Arquivos auxiliares" há um arquivo compactado (aux14.zip) que contém todos os arquivos de testes abertos (entradas e saídas esperadas).
- O laboratório é composto de 10 testes abertos e 10 testes fechados.
- O limite máximo será de 20 submissões.
- Acesse o sistema SuSy com seu RA (apenas números) e a senha que você utiliza para fazer acesso ao sistema da DAC.
- Você deve seguir as instruções de submissão descritas no enunciado.
- Serão considerados apenas os resultados da última submissão.
- A submissão de um programa sem uma função recursiva válida implementada será considerada tentativa de fraude.
- Esta tarefa tem peso 4.
- O prazo final para submissão é dia 11/12/2022 (domingo).