

**TRABALHO PRÁTICO****Observações:**

1. Comece a fazer este trabalho imediatamente. Você nunca terá tanto tempo para resolvê-lo quanto agora!
2. **Data de entrega:** 4 de novembro de 2022, até às **23:59 horas**, ou antes.
3. **Linguagem:** Você deve escrever o seu programa obrigatoriamente na linguagem de programação C padrão.
4. **Plataforma computacional:** Use a linguagem de programação C padrão e algum compilador bem conhecido como o gcc (<https://gcc.gnu.org/>).
5. **Documentação:**
  - Uma documentação “mínima” (possivelmente até cinco páginas) que explique as fases de projeto e implementação, incluindo uma breve descrição de como você resolveu o problema descrito abaixo, incluindo uma discussão sobre o projeto das estruturas de dados.
  - Um arquivo `leiametext`, a ser incluído no arquivo zip, com informações sobre o ambiente computacional para executar o seu TP, i.e., sistema operacional, bem como o compilador usado.
6. **Submissão:** Submeta o seu trabalho via Moodle, sendo um arquivo zip, descrito abaixo, com o nome `MD_ "SeuNomeCompleto".zip` onde o string "SeuNomeCompleto" é o seu nome completo sem espaços em branco.

Exemplo para o aluno Zoroastro Felizardo e Sortudo:

- Arquivo zip: `MD_ZoroastroFelizardoESortudo.zip`
- Conteúdo do arquivo zip: o seu arquivo zip deve ter **apenas** três arquivos:
  - (a) `leiametext`: arquivo texto como explicado acima.
  - (b) `passeio.c`: arquivo fonte na linguagem C para o passeio do cavalo.
  - (c) `passeio.pdf`: arquivo pdf da documentação de seu trabalho descrevendo as técnicas utilizadas para resolver este trabalho.

**Não inclua arquivos executáveis nesse arquivo zip!**

---

## Técnica de *backtracking* (“tentativa e erro”) e o Passeio do Cavalo

**Motivação para fazer este trabalho.** Uma área muito interessante de projeto de algoritmos é quando se quer achar soluções para problemas que não seguem uma **regra fixa** de computação, mas usando uma técnica de tentativa e erro.

Neste tipo de problema, a abordagem mais comum é decompor o processo em um número finito de tarefas parciais. Geralmente estas tarefas são expressas naturalmente em termos recursivos e devem ser exploradas de forma exaustiva. A construção de uma solução é obtida através de tentativas (ou pesquisas) da árvore de sub-tarefas. Muitas vezes estas tentativas crescem exponencialmente e, nestes casos, deve-se usar heurísticas para evitar uma pesquisa exaustiva. Uma heurística não garante a solução ótima mas tende a ser rápida. O objetivo aqui é de mostrar como esta técnica funciona e não heurísticas que, em geral, são dependentes do tipo de problema que se está estudando.

O “custo computacional” para resolver este tipo de problema depende exatamente de quantos “passos” são executados para se chegar a uma solução.

**Passeio do Cavalo.** A técnica de tentativa e erro pode ser melhor explicada através de um exemplo: O Passeio do Cavalo (Knight's Tour). Seja um tabuleiro  $n \times n$  com  $n^2$  posições e um cavalo que move seguindo as regras do xadrez. O cavalo é colocado numa posição inicial  $(x_0, y_0)$ . O objetivo do problema é encontrar, se existir, um passeio do cavalo com cada casa marcada com número que varia de 1 a  $n^2$ , sendo que a casa 1 é a inicial e cada número na sequência de 2 a  $n^2$  representa uma casa visitada de acordo com um movimento válido do cavalo.

Algoritmos que usam a técnica de tentativa e erro não seguem uma regra fixa de computação. Em geral, os passos em direção a solução final são tentados e registrados numa estrutura de dados. Caso esses passos tomados não levem para a solução final do problema, eles podem ser retirados e apagados do registro.

Neste problema, os oito movimentos possíveis de um cavalo estão representados a seguir, considerando a sua posição inicial identificada por **C** em um tabuleiro  $5 \times 5$ :

	$M_3$		$M_2$	
$M_4$				$M_1$
		<b>C</b>		
$M_5$				$M_8$
	$M_6$		$M_7$	

O movimento 7 ( $M_7$ ), por exemplo, significa avançar para a próxima coluna e descer duas linhas.

Uma possível solução para um tabuleiro  $5 \times 5$  seria:

23	10	15	4	25
16	5	24	9	14
11	22	1	18	3
6	17	20	13	8
21	12	7	2	19

Neste caso, o cavalo ao começar na posição indicada por 1 pode fazer qualquer um dos oito movimentos acima. O movimento 7 ( $M_7$ ) é executado e o número dois é colocado nessa posição.

Nesse momento, somente os movimentos 2 ( $M_2$ ) e 4 ( $M_4$ ) podem ser executados e um deles deve ser escolhido. Note que os movimentos  $M_1$  e  $M_5$  a  $M_8$  levam o cavalo para fora do tabuleiro, enquanto o movimento  $M_3$  retorna à posição inicial.

Para o exemplo acima, os movimentos que levam ao passeio das 25 casas do tabuleiro são:

Passo	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Movimento	7	2	3	5	6	8	1	2	4	6	7	1	2	4	5	7	1	7	4	5	2	3	8	1

Observe que esta é uma possível solução. Outras soluções existem. No entanto, qualquer solução que leve a um passeio usando uma “regra fixa”, ou seja, sem retrocesso (*backtracking*) deve ser capaz de “advinhar” de alguma forma esses passos. Como não se conhece essa regra fixa para um caso genérico, uma estratégia para resolver esse problema é a “tentativa e erro”.

Assim, na tentativa de buscar um passeio do cavalo, movimentos já feitos podem ser desfeitos, ou seja, deve-se “retroceder” (*backtracking*) até uma casa do tabuleiro onde um outro movimento pode ser escolhido e prosseguir com esse passeio. Por exemplo, no tabuleiro da figura abaixo à esquerda, a casa marcada com o número 24 é alcançada a partir da casa com o número 23 usando o movimento  $M_2$ . No entanto, a partir daí não é possível prosseguir. Isso significa que deve-se desfazer esse passo, escolher um outro movimento para uma outra casa, se for possível (caso não seja, outros retrocessos devem ocorrer). Por exemplo, a partir da casa com o número 23 e usando o movimento  $M_5$  iremos para uma outra casa marcada com o número 24 e podemos prosseguir, como mostrado na figura abaixo à direita.

1	8	5	16	3	10	13	24
	17	2	9	6	15		11
	20	7	4		12	23	14
18			21				
		19			22		

1	8	5	16	3	10	13	
	17	2	9	6	15		11
	20	7	4		12	23	14
18			21	24			
		19			22		

Esse cenário ilustra o processo de avançar e retroceder (*backtracking*). Como mencionado, pode ocorrer de vários passos terem que ser desfeitos dependendo das escolhas dos movimentos e da posição inicial do cavalo.

Para jogar o “Passeio do Cavalo” online, veja, por exemplo, <http://www.maths-resources.com/knights/>

**Entrada.** Não há entrada para este trabalho. Você deve apresentar o passeio do cavalo para um tabuleiro  $8 \times 8$ .

**Saída.** As seguintes saídas devem ser apresentadas:

1. Apresente um possível passeio do cavalo começando na posição mais acima e mais à esquerda do tabuleiro, i.e., posição (1, 1).
2. Apresente um possível passeio do cavalo começando em uma posição aleatória do tabuleiro diferente da posição (1, 1).
3. Para cada passeio indique a quantidade de casas visitadas e a quantidade de casas que tiveram que ser retrocedidas.

**Sugestão que não precisa ser entregue:** tente outras posições iniciais, outras sequências de movimentos e faça uma análise da quantidade de casas visitadas e quantidade de casas que tiveram que ser retrocedidas. Caso faça isso, apresente seus resultados na documentação do trabalho.

Os detalhes do formato da saída serão fornecidos assim que o monitor da disciplina tiver sido alocado.