Trabalho de Implementação para compor a nota 1 da disciplina IA:

* Data da entrega: 17 de julho de 2025, meia-noite.
* A turma será dividida em equipes de 5 ou 6 membros. **Não serão aceitas** equipes com quantidades diferentes dessas!
* Cada equipe deve selecionar 1 tema e 1 pessoa da equipe deve me enviar o nome do tema e os nomes dos participantes.
* Cada membro da equipe deverá assumir 1 dos seguintes papéis (pode haver mais de 1 pessoa assumindo o mesmo papel, mas não pode haver trocas de papéis ou múltiplos papéis assumidos no mesmo trabalho):
  + Criador e executor de testes de unidade para TDD (1 ou 2 pessoas);
  + Implementador do código (2 ou 3 pessoas);
  + Analista de Think Aloud (1 ou 2 pessoas).
* Será fornecido pelo professor um formulário para envio dos resultados.
* O que deve ser entregue:
  + Um documento pdf com 5 a 10 páginas. A primeira página deve conter o título do trabalho, os nomes dos participantes e seus números de matrícula e os papéis que cada um desempenhou no trabalho. As outras 9 páginas devem conter os resultados dos itens correspondentes: TDD+cobertura, Relatório Think Aloud, Testes Estáticos, Métricas Estatísticas.
  + Código zipado;
  + Testes zipados.

**Temas propostos para implementação**

1. Implementar jogo da velha com BFS [Gabriella Letícia, Kemilli Nicole, Júlia Leal, Oscar Torreão, João Limeira];
2. Implementar quebra-cabeça dos 8 números com 1 espaço vazio com Greedy BFS;
3. Implementar busca pela menor distância entre cidades usando algoritmo A\*. Um desafio que pode contar pontos extras se for implementado é usar prints do Google Maps como entrada, fazer o processamento da imagem e extrair o grafo com distâncias. Ou usar informações reais de distâncias entre cidades (Maria Luiza Galdino Medeiros; Guilherme Jose Araujo Oliveira; Rafael Alencar Adelino de Lima; Isaque Esdras Rocha da Silva Soares Cavalcante; Arthur Fernandes Vieira; Jefferson Ribeiro Brasil)
4. Implementar solução com heurística para o problema das 8 rainhas;
5. Implementar jogo da velha com busca adversarial;
6. Implementar busca pela menor distância entre cidades usando algoritmo de Dijkstra. Um desafio que pode contar pontos extras se for implementado é usar prints do Google Maps como entrada, fazer o processamento da imagem e extrair o grafo com distâncias. Ou usar informações reais de distâncias entre cidades; (João Matheus Pinto Villarim, Coutinho de Almeida, Aelson Júnior Araújo Diniz da Cunha, Beatriz de Sousa Meneses, Rafael Barreto da Silva, Matheus Galdino de Souza, Vinicius de Oliveira Porto)
7. Treinar um algoritmo de Máquina de Vetores de Suporte utilizando kernel rbf (seguindo este tutorial: [link](https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html)). No tutorial, a busca pelos parâmetros c e g é feita por grid-search. Utilize um algoritmo genético (genetic-search) simples para fazer a busca pelos parâmetros c e g. Compare os resultados da grid-search com a genetic-search.
8. Treinar uma rede neural artificial MLP com apenas 1 camada escondida para classificar as 3 categorias de flores da base de dados iris-dataset. Utilize um algoritmo genético simples para fazer a busca pelos melhores parâmetros de treinamento da rede. Faça outra sequência de treinamentos fazendo a busca dos parâmetros escolhendo valores aleatoriamente a partir de uma faixa predefinida de valores (Alexandre Tostes Campos , Jonathas Alves Jorge, Vinicius Gabriel Laureano, Vitor Schuler Velloso Borges, Wendel Silva Italiano de Araújo).
9. Desenvolva um algoritmo genético simples para solucionar o seguinte problema:
   * **Encontre os valores de entrada que maximizam ou minimizam uma dada função matemática**, como \( f(x)=x^{2}\) ou \(f(x,y)=(x-2)^{2}+(y-3)^{2} \)[[1]](#footnote-0).
   * **Representação**: Cada indivíduo na população representa um conjunto de valores de entrada (por exemplo, uma lista de números de ponto flutuante).
   * **Função de Aptidão**: A função a ser otimizada (por exemplo, \(f(x,y)\)).
   * **Operações Genéticas**: **Seleção**: Indivíduos com maior aptidão (menor valor para um problema de minimização) têm maior probabilidade de serem selecionados como pais. **Cruzamento**: Os cromossomos dos pais são combinados para criar novos descendentes. Por exemplo, com dois cromossomos dos pais e um cruzamento pode produzir descendentes ou. **Mutação**: Mudanças aleatórias são introduzidas nos cromossomos dos descendentes (por exemplo, alterando ligeiramente um número de ponto flutuante). [Gustavo Nogueira, Lenner Coutinho, Pedro Henriques, Arthur Vidal, Nivea Calebia]
10. Desenvolva um algoritmo genético simples para solucionar o seguinte problema: Auxiliar um "Carro" a Percorrer um Percurso (José Ramon Severo Alves, Francisco Pereira da Silva Filho, Geraldo de Lima Junior, Cristian Alves da Silva, Carlos Henriche, Abner de Lucena Leão Gonçalves, Lucas Oliveira Carvalho) :
    * **Problema**: Projete um "carro" (representado por um conjunto de parâmetros) que possa percorrer um percurso simples em 2D ou 3D, talvez com obstáculos.
    * **Representação**: O "carro" pode ser representado por um conjunto de parâmetros, como tamanho da roda, potência do motor, etc.
    * **Função de Aptidão**: A distância percorrida ou o tempo gasto para completar o percurso.
    * **Operações Genéticas**: Semelhante à otimização da função matemática, mas a função de aptidão é mais complexa e pode envolver a simulação do movimento do carro.
11. Desenvolva um algoritmo genético simples para solucionar o seguinte problema: **"Problema dos 100 Prisioneiros"**:
    * **Problema**: Um grupo de 100 prisioneiros recebe, cada um, uma caixa numerada. Uma caixa contém uma chave para sua liberdade, e os prisioneiros só podem olhar em um número limitado de caixas (por exemplo, 50) antes de decidir se querem escapar. O objetivo é encontrar uma estratégia (uma sequência de aberturas de caixas) que lhes dê uma alta probabilidade de sucesso.
    * **Representação**: Cada indivíduo representa a estratégia de um prisioneiro (uma sequência de números de caixas para abrir).
    * **Função de Aptidão**: A probabilidade de sucesso (encontrar a chave) com base na estratégia escolhida.
    * **Operações Genéticas**: Seleção padrão, cruzamento e mutação são usados ​​para desenvolver estratégias.

1. As equações aqui neste texto estão escritas usando notação LaTex. [↑](#footnote-ref-0)