

Disciplina: Paradigmas de Programação
Professor: Maicon Rafael Zatelli
Entrega: Moodle

Atividade V - Prolog

Atenção: Faça um ZIP com todos os arquivos de solução. Use o nome do arquivo de maneira a entender qual problema você está resolvendo. Por exemplo, problema1.pl, problema2.pl e assim por diante.

- Crie exemplos de consultas para testar sua solução e adicione em comentário a consulta efetuada e a solução dada pelo Prolog.

1. Crie uma estrutura de dados para pontos 2D.

- A:** Crie uma regra `distancia`, a qual recebe dois pontos como parâmetro e deve retornar a distância entre eles.
- B:** Crie uma regra `colineares`, a qual recebe três pontos como parâmetro e é verdade se eles são colineares. DICA: os três pontos serão colineares se o determinante de suas coordenadas for igual a 0.
- C:** Crie uma regra `formaTriangulo`, a qual recebe três pontos como parâmetro e é verdade se eles podem ser usados para formar um triângulo.
- D:** Faça consultas para demonstrar o uso e inclua (em comentário) no final do seu arquivo `.pl`.

2. Modifique o arquivo `arvore.pl` (disponível no Moodle) de forma a adicionar novas operações a nossa árvore. Assuma que nossa árvore não é uma árvore binária de busca.

- A:** Crie uma regra que receba um valor X e uma árvore e retorne a quantidade de ocorrências dele na árvore.
- B:** Crie uma regra que receba um valor X e uma árvore e retorne a quantidade de elementos maiores que ele na árvore.
- C:** Crie uma regra que receba uma árvore e retorne a média dos elementos na árvore.
- D:** Crie uma regra que receba uma árvore e retorne a quantidade de elementos na árvore.
- E:** Crie uma regra para transformar uma árvore em uma lista. Esta lista deve simplesmente conter todos os elementos da árvore.
- F:** Crie uma regra que receba um valor X e uma árvore e seja verdade quando o valor X não está presente na árvore.
- G:** Crie uma regra que receba uma árvore e retorne a lista de números visitados considerando o percurso em pós-ordem.
- H:** Crie uma regra que receba uma árvore e retorne a lista de números visitados considerando o percurso em pré-ordem.
- I:** Crie uma regra que receba uma árvore e retorne a lista de números visitados considerando o percurso em em-ordem.
- J:** Crie uma regra que receba uma árvore e retorne a quantidade de folhas nela existentes.
- K:** Crie uma regra que receba uma árvore e retorne uma lista com todos os elementos encontrados nas folhas da árvore.
- L:** Crie uma regra que receba duas árvores e seja verdade quando as duas árvores forem iguais. Duas árvores são iguais se elas possuem os mesmos elementos, dispostos da mesma forma.
- M:** Crie uma regra que receba uma árvore e retorne a altura da árvore. Uma árvore com apenas o nó raiz possui altura 0. Assuma que não há nenhuma árvore com nenhum nó.
- N:** Crie uma regra que receba uma árvore e dois elementos, X e Y, (sempre presentes na árvore) e retorne o caminho (uma lista de elementos) entre os dois elementos, ou seja, que nós devem ser visitados para ir de um nó com o elemento X até um nó com o elemento Y.
- O:** Faça consultas para demonstrar o uso de cada item e inclua (em comentário) no final do seu arquivo `.pl`.