Disciplina: Paradigmas de Programação
Professor: Maicon Rafael Zatelli
Entrega: *Moodle* (**basta um membro do grupo entregar**)

Trabalho I - Programação Funcional - Haskell

Atenção: este trabalho poderá ser feito em grupos de até **3 pessoas**.

Descrição

Neste trabalho, seu grupo deverá criar um resolvedor para um dos puzzles abaixo (**escolha um deles**), na linguagem **Haskell**.

`https://www.janko.at/Raetsel/Kojun/index.htm`

ou

`https://www.janko.at/Raetsel/Makaro/index.htm`
ou

`https://www.janko.at/Raetsel/Sudoku/Vergleich/index.htm`

Não preocupe-se com o desempenho da sua solução, e foque em tamanhos de tabuleiros de até 10x10 no primeiro e segundo puzzle, e até 9x9 no terceiro puzzle. Só depois tente melhorar o código para também satisfazer tabuleiros maiores. Além disso, implemente a entrada e a saída (resposta do programa) da forma que o grupo considerar melhor. A entrada, por exemplo, pode ser fornecida diretamente no código fonte, sem que seja necessária a digitação por parte do usuário.

Dica 1: a técnica de programação mais adequada para resolver este problema é a da "tentativa e erro" (*backtracking*). Pesquise como utilizá-la em Haskell.

Dica 2: **aprenda a jogar o puzzle** e depois pense em como modelar o problema (em especial, o tabuleiro em si) por meio de alguma estrutura de dados adequada (ex: matriz, árvore, etc). Se preferir, primeiro resolva o puzzle em alguma linguagem que o grupo domina (ex: Python, C++, Java).

Dica 3: a página pode ser traduzida para inglês ou português, para melhor compreensão das regras. Você também pode acessar esta página (`https://www.janko.at/Raetsel/Rules.htm`) para entender o que os autores dos puzzles querem dizer com algumas palavras, como "area", "orthogonally", "diagonal", "region", etc.

Dica 4: procure como criar um resolvedor do puzzle Sudoku em Haskell, aprenda como funciona e faça as adaptações necessárias para resolver o problema proposto para este trabalho.

Entregas e Apresentação

Os seguintes itens devem ser entregues:

- (**70% da pontuação**) Código fonte da solução **comentado**
- (**30% da pontuação**) Apresentação

Apresentação

O trabalho será apresentado para o professor em sala de aula.

- A apresentação deverá ilustrar o funcionamento da solução, isto é, o grupo deve executar a solução desenvolvida considerando algum exemplo de tabuleiro disponível na página do puzzle.
- O grupo deverá explicar brevemente os trechos de código que considerar importante (ex: modelagem/implementação do tabuleiro, algumas funções criadas, otimizações empregadas, etc).
- É esperado que a apresentação tenha a duração máxima de 10 minutos.
- A apresentação do trabalho é obrigatória para receber qualquer nota, ou seja, **trabalhos não apresentados terão nota 0**.

Avaliação

Após a avaliação das entregas, o grupo receberá como resultado uma pontuação proporcional ao número de membros do grupo, ou seja, suponha que o número de membros do grupo é N, então a pontuação dada pelo professor ao grupo será um valor entre 0 e N*10. Após receber esta pontuação, o grupo deverá fazer uma auto-avaliação e dividir a pontuação entre todos os membros do grupo de forma que a pontuação máxima para um determinado membro do grupo não ultrapasse 10 e não seja fracionada aquém ou além de 0,5. Cada grupo pode decidir os critérios que considerar melhor para dividir a pontuação. O grupo deve entregar ao professor o resultado de como a pontuação ficou dividida entre os membros (nome, matrícula e pontuação de cada membro), para que o professor possa efetivar a nota de cada membro do grupo no trabalho.

Note que, mesmo seguindo as dicas, há várias formas de resolver este problema. Assim, se for constatado cópia da solução, ambos os grupos (que copiou e o grupo que deixou copiar) levarão nota zero.