

1 Instruções Importantes

Nessa seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem funcional Haskell.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- relatório elucidando como o problema foi modelado e solucionado, em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código executa no *ghci*. Programas com erros de sintaxe receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo do relatório tenha a identificação do autor do trabalho;
4. arquivo compactado com os artefatos estão devidamente identificados com nome e matrícula.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, relatório e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

O relatório do trabalho deve conter informações relevantes para esclarecer a estratégia intelectual utilizada para modelar e resolver o tema do trabalho. Deve-se apresentar as decisões de projetos tomadas, em especial, as estruturas de dados para modelar e solucionar o problema.

O trabalho deverá ser apresentado ao professor da disciplina e, só será avaliado após a realização da entrevista, i.e., trabalhos que não forem apresentados não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, relatório e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina.**

Atrasos serão penalizados por uma função exponencial de dias de atraso, i.e., será reduzido da nota um percentual referente a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

Dias de Atraso	Nota
1	$n \cdot 0.98$
2	$n \cdot 0.96$
3	$n \cdot 0.92$
4	$n \cdot 0.84$
5	$n \cdot 0.68$
6	$n \cdot 0.36$
7	0

2 O Jogo Nim

Nim é um jogo de raciocínio jogado com “palitos” por dois jogadores. O jogo consiste em quatro fileiras com palitos. A primeira fileira contém 1 palito, a segunda 3 palitos, a terceira 5 palitos e a quarta 7 palitos, conforme ilustrado na Figura 1.

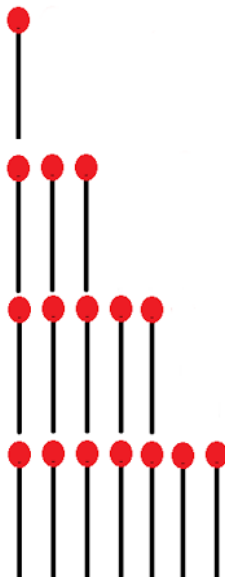


Figura 1: Situação inicial do jogo Nim.

A cada rodada, um jogador escolhe uma fileira e remove de 1 até todos os palitos da fileira e passa a vez para o próximo jogador. O jogo prossegue nesta dinâmica até que um jogador remova o último palito, sendo declarado o vencedor.

3 Especificação Técnica do Trabalho

O objetivo deste trabalho é implementar, em Haskell, o jogo *Nim*, com o usuário jogando contra o “computador”. A cada rodada, o programa exibirá no terminal as linhas com a respectiva quantidade de palitos e aguardará que o jogador informe a quantidade de palitos que deseja remover de uma determinada linha. Após obter a jogada do jogador, o programa deverá realizá-la e, em seguida, realizar a jogada do

computador e exibir a nova configuração do jogo. O programa repetirá este processo até que haja um vencedor, exibindo o resultado final.

Deverá ser implementado dois níveis de “inteligência” para o computador. No nível *fácil*, o computador irá selecionar uma quantidade aleatória para remover de palitos de uma determinada linha. Neste nível, o jogo será iniciado pelo jogador. No nível *difícil*, o computador deverá fazer sempre a jogada perfeita, conforme a estratégia vencedora descrita na Seção 3.1. Neste nível, o computador começará a jogada.

3.1 Estratégia Vencedora do Jogo Nim

O jogo *Nim* tem uma estratégia vencedora. Para tal, deve-se representar a quantidade de palitos de cada fileira na base 2. Por exemplo, suponha que tenham 1, 3, 4 e 7 palitos em cada uma das fileiras. A primeira etapa da estratégia vencedora é representar a quantidade de palitos das fileiras na base 2, que neste exemplo será 001, 011, 010 e 111, respectivamente. A próxima etapa é realizar a soma destas representações:

$$\begin{array}{r} 0 \ 0 \ 1 \\ 0 \ 1 \ 1 \\ 0 \ 1 \ 0 \\ 1 \ 1 \ 1 \\ \hline 1 \ 3 \ 3 \end{array}$$

Observe que foi feita a soma decimal da representação binária e o resultado foi 133. A estratégia vencedora consiste em sempre remover uma quantidade de palitos de modo que o resultado da soma tenha todos os algarismos pares. No exemplo acima, se for removido todos os 7 palitos da última fileira a soma resultante terá somente algarismos pares:

$$\begin{array}{r} 0 \ 0 \ 1 \\ 0 \ 1 \ 1 \\ 0 \ 1 \ 0 \\ 0 \ 0 \ 0 \\ \hline 0 \ 2 \ 2 \end{array}$$

Este procedimento deve ser realizado a cada jogada, garantindo a vitória do primeiro jogador que conseguir deixar o resultado da soma das representações binárias com algarismos pares.

4 Entrega do Trabalho

A data da entrega do trabalho será até o dia **14 de janeiro de 2022**, via plataforma do GoogleClassroom.

5 Bônus

Os trabalhos que implementarem também uma versão gráfica do jogo serão bonificados com a pontuação extra de 5 pontos. Aos interessados em implementar um modo gráfico, recomendo a biblioteca *gtk2hs*.