

Fundamentos da Programação - Projeto 4

Autores: Gabriel Franco e Dahan Schuster

Sobre a contribuição dos integrantes

Excluindo as duas primeiras que eram mais fáceis e houve um consenso imediato de como resolvê-las, todas as funções geraram discussão entre a dupla, especialmente as duas últimas questões. Como o projeto foi feito no Replit ambos da equipe contribuíram para o código, nós trocamos ideias de como resolver certos problemas enquanto alterávamos o programa e testávamos novas soluções.

Detalhes extras sobre as funções

Terceira função: **hardClipping**

Nesta função tivemos que pensar numa forma de não atribuir o valor positivo do limite caso a amostra seja negativa, ou seja, esteja em um vale. O **if** mais interno é responsável por verificar se o limite e a amostra possuem sinais iguais. Caso positivo, a amostra recebe o limite. Caso contrário, a amostra deve receber o valor oposto (multiplicado por -1)

Sobre os desafios

Até a terceira função, podemos dizer que os desafios se pareciam muito com outras questões envolvendo vetores. Porém na função **limitaSinal** sentimos uma maior dificuldade e tivemos que pensar numa estratégia mais sofisticada. O fato do contexto para o algoritmo ser totalmente diferente dos vistos até então deixou o desafio mais interessante.

Para fazer a **limitaSinal**, a mais difícil na visão do Gabriel, tivemos que testar diferentes aspectos do algoritmo, isto é, como ele deveria se comportar em situações diferentes. Este talvez foi o tópico que mais contribuiu pra chegarmos a uma versão final, pois a cada teste falho nós percebíamos algum erro e melhorávamos a função. Mesmo depois da função pronta, ainda ficamos em dúvida se estava correta ou não, pois o resultado parecia estar errado mesmo que o algoritmo estivesse correto. Não entendíamos porque o pico mais alto passava a ser o mais a direita, ou porque a curva da onda não continuava proporcionalmente a mesma. Por exemplo:

- Vetor entrada: { 2.00, 2.00, 2.00, 3.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00 }
- Vetor saída: { 0.63, 0.59, 0.55, 0.75, 0.58, 0.62, 0.66, 0.73, 0.85, 1.00 }

Pode perceber que, enquanto a curva inicial é uma constante de valor 2 com um pico no valor 3, a saída é uma curva com mais nuances. Essa dúvida só foi tirada depois que conversamos com o Bogdan sobre o desafio.

A função `geraOndaQuadrada`, a mais difícil na visão do Dahan, também exigiu bastante criatividade pra resolver. Talvez a parte mais difícil tenha sido entender de onde tirar cada informação para encontrar o caminho a se percorrer no algoritmo, como as variáveis `periodo`, `meio_periodo` e `ciclos`. Tendo essas informações, o desafio foi saber como exatamente deveria funcionar a função, pois o conceito de meio-período e quantidade de ciclos deixou tudo meio confuso.

Sobre a superação dos desafios

A solução para todos os desafios deste projeto foi pensada em conjunto enquanto discutíamos possíveis soluções. Apesar de um ou outro ter chegado na solução primeiro, isso só foi possível porque ambos estavam trocando o que achava ser útil sobre o problema, e escrevendo junto.