

Trabalho 05 - Um Jogo muito Nerd

Data de entrega: 19/12/2022 até 20:59, peso 1/2

Importante:

- **Não** olhe códigos de outros ou da internet. Exceto os que são fornecidos. Também não mostre ou publique o seu.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes.

Este trabalho é em dupla com as mesmas duplas registradas e deverá ser implementado em linguagem C.

Considere o seguinte jogo, um mestre irá te falar uma sequência de números. A cada número que ele falar, você precisa responder dizendo a mediana dos números que foram dados até aquele momento. Considere o seguinte exemplo:

Mestre: 5

Você: Mediana: 5.0

Mestre: 10

Você: Mediana: 7.5

Mestre: 1

Você: Mediana: 5

O jogo continua até o mestre passar o número -1 (um negativo) e tendo acertado todas as medianas você ganha o jogo. Você então decide fazer um programa, uma ideia bastante óbvia é manter um vértice ordenado, e assim que um novo número é dito pelo mestre você coloca no vetor, e pega a mediana.

Porém essa solução começa a ficar muito custosa quando a quantidade de números começa a crescer demais. Então você precisa de uma solução mais rápida.

Ideia: Manter 2 conjuntos, um com os maiores elementos, e outro com os menores elementos, de forma que cada um tem metade dos números originais (se houver um número ímpar de elementos, um desses conjuntos tem 1 elemento a mais).

Nessa solução quando você vai encontrar a mediana você precisa encontrar o menor do conjunto dos Maiores elementos, e o maior do conjunto dos Menores elementos. E por isso parece uma boa solução você implementar esses conjuntos como Heaps!

O conjunto dos menores elementos deve utilizar um heap de máximo (Max Heap), enquanto o conjunto dos maiores elementos deve utilizar um heap de mínimo (Min Heap). Você receberá uma implementação MaxHeap.c e seu trabalho é implementar o MinHeap.c e submeter apenas esse arquivo.