BigNumber

Nomes e RA's:
11201921774 - Guilherme Ferreira Costa
11202020632 - Henrique Allucci Gonçalves

– Uma explicação de como foi representado o BigNumber.

```
typedef struct {
        int *v_numbers;
        int tam;
        int sinal; // 1-pos | 0-neg
} Bignumber;
```

Refatoramos a representação do nosso BigNumber utilizando uma estrutura (struct) em C. Os parâmetros da struct incluem:

- Um vetor de inteiros (int *v_numbers), que é gerado a partir de uma string.
- O tamanho do vetor (int tam).
- O sinal do BigNumber (int sinal), onde 1 representa positivo e 0 representa negativo.

Essa abordagem mais organizada e estruturada permite uma manipulação mais eficiente e clara do BigNumber no contexto do programa. Além disso, facilita a leitura e compreensão do código por parte de desenvolvedores.

– Qual a interface pública do seu tipo BigNumber (basta as assinaturas).
As funções utilizadas foram as seguintes:

```
Bignumber criar_bignumber(char *number);
void imprimir_bignumber(Bignumber* bigNum);
void free_bignumber(Bignumber *bignumber);
void soma(Bignumber *a, Bignumber *b);
void subtracao(Bignumber *a, Bignumber *b, int maior, char operacao);
int maior_num(Bignumber *a, Bignumber *b);
```

– Mencione qualquer algoritmo ou estrutura de dados avançada que tenha sido empregada para melhorar o tempo de execução do seu código. Além disso, diga como/onde usou.
Com o objetivo de otimizar o tempo de execução, aprimoramos o código ao modificar o maior número entre os dois fornecidos na entrada, eliminando assim a necessidade de criar um terceiro número que alocasse mais memória no computador, contribuindo para uma execução mais eficiente. Além disso, otimizamos os parâmetros das funções, passando apenas os ponteiros dos valores, evitando assim cópias desnecessárias do número fornecido. Essas modificações foram implementadas para otimizar o desempenho do programa.

- Divisão de trabalho:
Para otimizar nosso processo de desenvolvimento, dividimos as tarefas iniciais entre dois colaboradores. Uma pessoa ficou encarregada da implementação inicial da função de soma, enquanto a outra pessoa assumiu a responsabilidade pela subtração.
Posteriormente, unimos esforços para realizar uma depuração conjunta do código. Durante essa fase, trabalhamos em equipe para identificar e corrigir problemas que surgiram durante os testes