

`big_val` -> Começamos o trabalho com a função, pois não poderíamos implementar qualquer outra sem ela. A função funciona da maneira esperada e não mostra nenhum erro de acordo com os nossos testes.

Testes:

- Os testes são essencialmente usados nas outras funções, pois usamos a função para passar valores em algumas outras funções, como `big_sum`, `big_sub`, `big_comp2` e `big_mul`.
- Criamos a função `print_val` para imprimir o valor inteiro, em hexadecimal do `BigInt`, mostrando em cada teste que funciona da maneira esperada.

`big_comp2` -> Funciona da maneira esperada, também utilizada na função de subtração. Pensamos em usar na função de multiplicação, mas percebemos que não é necessário, logo permanece sendo vital utilizá-la somente na função de subtração. Fizemos testes separados, mas também foi indiretamente testada nas funções de subtração, pelo motivo mencionado anteriormente.

`big_sum` -> Funciona da maneira esperada, testado nos seguintes casos com números: positivo e negativo, negativo e negativo, positivo com positivo.

`big_sub` -> Mesmos casos de teste do que `big_sum` (números diferentes).

`big_mul` -> Mesmos casos de testes do que `big_sum` e `big_sub` (números diferentes). Algoritmo que mais demorou tempo para ser desenvolvido, por causa de vários possíveis overflows, durante a multiplicação em si ou durante a soma de multiplicações.

`big_shl` -> Funciona da maneira esperada. Testada com os maiores e menores valores possíveis no `BigInt`, como 1 e `0xffff...ffff`, também testado com valores de shift múltiplos de 8, para ver se falhava em alguma passagem de overflow. Foi a função mais demorada para fazer dentre os shifts, já que foi a primeira a ser elaborada.

`big_shr` -> Funciona da maneira esperada. Testada da mesma maneira que `big_shl`.

`big_sar` -> Funciona da maneira esperada. Testada da mesma maneira que `big_shl`, porém tomando cuidado para verificar os maiores e menores valores positivos e negativos do `BigInt`. Também comparamos com a função `big_shr`, que já havia sido testada nos casos de números positivos (unsigned).