

Prova de Suficiência

23 de abril de 2018

1 \mathbb{N}^* orientado a objetos

(a) (1,25)

Implemente um objeto capaz de representar um elemento de \mathbb{N}^* (números naturais maiores que zero). O objeto deve obedecer à seguinte interface, e às seguintes restrições de complexidade assintótica para cada operação:

- `Numero um()` — $\Theta(1)$
Constrói o número 1.
- `Numero soma(Numero self, Numero outro)` — $\Theta(1)$
Constrói o número `self + outro`.
- `repetir(Numero self, Funcao funcao)` — $\Theta(\text{self}) \times (\text{complexidade da função})$
Chama `self` vezes uma função.

(b) (1,25)

Implemente uma rotina capaz de construir um objeto `Numero` a partir qualquer número N (um inteiro nativo da sua linguagem de programação favorita, por exemplo um `int`) utilizando apenas os construtores acima (`um` e `soma`), no melhor tempo possível e com o melhor consumo de memória possível. Calcule o tempo de execução e o consumo de memória da sua rotina em notação assintótica.

2 (7,5) *Contest*

Participe do *contest* em <https://www.codepit.io/#/contest/5addb2d1636fa800962e7f27/view> (senha: rjuapboa) e entregue suas soluções. Cada problema vale 2,5. Seus códigos devem ser originais — qualquer solução plagiada receberá a nota zero.