

Variação Bubble

Um estudante de programação decidiu experimentar algumas variações do bubblesort. Uma das idéias que teve foi a de comparar e trocar, se necessário, não elementos consecutivos do vetor, mas elementos simétricos de uma parte do vetor. Por exemplo, se considerar o vetor inteiro, devem ser comparados o primeiro e o último elementos do vetor; em seguida, o segundo e o penúltimo elementos; e assim por diante.

Pede-se que você implemente a função `variacao_bubble(vetor, ini, fim)`

Semântica da função

A função deve operar sobre uma lista, tratando-a como um vetor. Ou seja, nenhuma operação de alteração estrutural da lista deve ser realizada, tais como inserções, remoções, slices, etc. Apenas a atribuição de valores é possível. Observe que a função também deve retornar o número de trocas realizadas.

```
vetor = [6, 9, 2, 3, 1, 4]
assert variacao_bubble(vetor, 0, 6) == 2
assert vetor == [4, 1, 2, 3, 9, 6]
```

Os parâmetros `ini` e `fim` indicam os limites, usando a semântica tipicamente usada em slices de python. Ou seja, o parâmetro `ini` indica o índice inicial e o parâmetro `fim` indica o índice subsequente ao fim do trecho. Observe que a parte do vetor que fica fora dos limites indicados pelos parâmetros `ini` e `fim` não deve ser alterada.

```
vetor = [8, 6, 9, 2, 3, 1, 4, 0, 7, 4]
assert variacao_bubble(vetor, 1, 7) == 2
assert vetor == [8, 4, 1, 2, 3, 9, 6, 0, 7, 4]
```

Última atualização por daltonserey, 2 anos atrás

Para enviar sua resposta:

1. faça o *checkout* da atividade com o comando `tst checkout 5252841717366784`;
2. faça a atividade, criando sua resposta em um arquivo no diretório do *checkout*;
3. faça o *commit* com o comando `tst commit {nome-do-seu-arquivo}`.