

Operação min_heapify

Escreva uma função, chamada min_heapify, que recebe como entrada um vetor contendo uma “quase-heap” mínima e transforma esse vetor em uma heap mínima.

Uma “quase-heap” mínima é um vetor $A = (a_1, a_2, a_3, \dots, a_{n-1})$ no qual todos os elementos respeitam a propriedade de heap mínima, com exceção do primeiro, que pode ou não respeitar a propriedade de heap mínima. Para transformar essa “quase-heap” em uma heap, compare o elemento na posição a ser testada do vetor com os seus dois filhos (se houver). Caso ele viole a propriedade de heap com algum dos filhos, troque-o de posição com o filho mais promissor. Aplique min_heapify recursivamente até que a propriedade de heap não seja mais violada.

Para recordar, dado um elemento $A[i]$ em uma heap...

- O pai de $A[i]$ pode ser encontrado no índice $\text{pai}(i) = \lfloor i/2 \rfloor$;
- O elemento $A[0]$ não possui pai;
- O filho esquerdo de $A[i]$, se existir, pode ser encontrado no índice $2i+1$ e o filho direito, se existir, em $2i+2$;
- A propriedade de heap mínima é $A[\text{pai}(i)] \leq A[i]$.

Em seguida, faça um programa principal que lê várias listas em notação Python, representando “quase-heaps”. Para cada uma imprima a lista resultante da operação min_heapify. A entrada termina com a lista vazia ([]).

Exemplos de Entrada e Saída

Entrada	[2, 6, 7] [3, 5, 4] [6, 7, 2] [5, 3, 1, 6, 4, 4, 8, 7, 7, 4] []
Saída	[2, 6, 7] [3, 5, 4] [2, 7, 6] [1, 3, 4, 6, 4, 5, 8, 7, 7, 4]