

Projeto e Análise de Algoritmos - Lista 1

FGV - EMAP

Setembro de 2025

Para os próximos exercícios você deve alterar o arquivo `lista-2-paa.py` adicionando a solução no corpo da função correspondente.

Atenção:

- Não altere o nome do arquivo Python.
- Para cada questão abaixo existe uma função definida no arquivo Python.
- Não altere a assinatura das funções. Você deve receber e retornar os tipos definidos.
- Caso queira definir uma classe ou função auxiliar, faça isso dentro da função do problema que estiver resolvendo.

1. A Biblioteca de Alexandria

A administração da biblioteca busca compreender o padrão de uso de seus frequentadores e, para isso, precisa identificar o período de pico, ou seja, o intervalo de tempo em que há o maior número de usuários presentes. A biblioteca dispõe de uma catraca que registra os horários de entrada e saída de cada pessoa.

Seu desafio é desenvolver um algoritmo com tempo de execução $O(n \log n)$ que resolva essa questão. O algoritmo deverá receber n pares de inteiros (a, b) , onde a representa o horário de entrada e b o de saída de um usuário. Como resultado, ele deve retornar $k, (u, v)$, onde k é a quantidade máxima de pessoas na biblioteca e (u, v) é o intervalo de tempo maximal (de maior tamanho) correspondente a esse pico. Se houver mais de um intervalo de pico maximal, retorne aquele com menor u .

Restrições:

- $1 \leq n \leq 2 \cdot 10^5$.
- $1 \leq a_i < b_i \leq 10^9$ para todo $1 \leq i \leq n$.
- $a_i \neq a_j, a_i \neq b_j, b_i \neq b_j$ para todo $1 \leq i < j \leq n$. Ou seja, todos os tempos de entrada e saída são distintos.

Entrada	Saída
$[(5, 8), (2, 4), (3, 9)]$	$2, (5, 8)$
$[(5, 8), (2, 6), (3, 9)]$	$3, (5, 6)$

2. Sisi e a Sorveteria

Sisi é uma gata que adora sorvete. Todos os dias, ela vai à sorveteria e pede o ‘sabor do dia’. Um belo dia, uma dúvida surgiu em sua mente: “Qual foi o período mais longo, em dias consecutivos, que passei sem repetir um único sabor?”.

Por sorte, ela possui todos os sorvetes que pediu anotados em seu diário, desde o primeiro dia.

Ajude-a com essa dúvida. Você deve desenvolver um algoritmo com tempo de execução $O(n)$ que receba a sequência de sabores consumidos e retorne o tamanho da maior subsequência contínua de valores distintos.

Entrada	Saída
[1, 2, 1, 4]	3
[11, 22, 11, 33, 22, 77, 44, 22]	5
[3, 3, 3, 3, 3, 3]	1

3. Hotel de Hilbert

O Grande Hotel de Hilbert está se preparando para sua reinauguração e receberá n hóspedes. Para cada hóspede, conhecemos um par de inteiros (a, b) , que representam seu tempo de chegada e de partida, respectivamente. Para minimizar os custos, o gerente deseja utilizar o menor número possível de quartos.

Sua tarefa é ajudar o gerente a distribuir os hóspedes nos quartos de forma otimizada. Duas pessoas podem ocupar o mesmo quarto, desde que o período de estadia delas não se sobreponha. Ou seja, uma pessoa pode usar um quarto após a outra se o tempo de saída da primeira for estritamente menor ao tempo de entrada da segunda. Os quartos são numerados de 1 a n .

Desenvolva um algoritmo com tempo de execução $O(n \log n)$ que resolva esse problema. O algoritmo deverá receber n pares de inteiros (a, b) , onde a representa o horário de entrada e b o de saída de um hóspede. Como resultado, ele deve retornar $k, [r_1, r_2, \dots, r_n]$, onde k é a quantidade mínima de quartos necessários e r_i é o quarto que o i -ésimo hóspede (**na mesma ordem da vetor de entrada**) deverá utilizar.

Caso exista mais de uma distribuição de quartos válida que utilize o número mínimo k , qualquer uma delas é uma resposta válida.

Restrições:

- $1 \leq n \leq 2 \cdot 10^5$.
- $1 \leq a_i \leq b_i \leq 10^9$ para todo $1 \leq i \leq n$.
- Para a saída: $1 \leq r_i \leq k$ para todo $1 \leq i \leq n$.

Entrada	Saída
[(1, 2), (3, 3), (4, 5)]	1, [1, 1, 1]
[(2, 4), (1, 2), (4, 4)]	2, [1, 2, 2] ou 2, [2, 1, 1]
[(1, 2), (2, 4), (4, 4)]	2, [1, 2, 1] ou 2, [2, 1, 2]
[(1, 4), (1, 2), (2, 4)]	3, [1, 2, 3] ou 3, [*qualquer permutação de 1, 2, 3*]

4. Quadra

Dado um vetor A com n inteiros e um valor alvo k , seu desafio é projetar um algoritmo que encontre quatro índices distintos cuja soma dos elementos seja igual a k .

O algoritmo deve ter uma complexidade de tempo de $O(n^2 \log n)$ e retornar uma tupla com os quatro índices em ordem crescente, (a, b, c, d) com $a < b < c < d$, que satisfaça a condição $A_a + A_b + A_c + A_d = k$.

Caso existam múltiplas soluções, retornar qualquer uma delas é suficiente. Se nenhuma combinação válida for encontrada, o algoritmo deve retornar $(-1, -1, -1, -1)$.

Restrições:

- $1 \leq n \leq 10^3$
- $1 \leq A_i \leq 10^9$ para todo $1 \leq i \leq n$
- $1 \leq k \leq 4 \cdot 10^9$

Exemplos:

Entrada	Saída
$A = [3, 2, 6, 1]$ e $k = 7$	$(-1, -1, -1, -1)$
$A = [3, 2, 6, 1]$ e $k = 12$	$(1, 2, 3, 4)$
$A = [3, 2, 5, 8, 1, 3, 2, 3]$ e $k = 15$	$(2, 4, 6, 7)$

5. Os blocos

Você recebeu n blocos de madeira e seu desafio é empilhá-los, formando o menor número possível de torres. Para isso, você deve seguir duas regras estritas:

- **Regra de Empilhamento:** Um bloco só pode ser colocado sobre outro se o seu tamanho for menor ou igual ao do bloco inferior.
- **Ordem de Processamento:** Os blocos devem ser processados um a um, na sequência predefinida em que são apresentados.

A cada bloco, você deve decidir se o coloca no topo de uma torre existente (respeitando a regra de empilhamento) ou se inicia uma nova torre com ele.

Dada a sequência de tamanhos dos blocos, encontre o número mínimo de torres necessárias. Desenvolva um algoritmo que execute em $O(n \log n)$ no pior caso, onde n é o número de blocos.

Entrada	Saída	Explicação
$[1, 2, 1, 4]$	3	$[1, 1], [2], [4]$
$[11, 22, 11, 33, 22, 77, 44, 22]$	4	$[11, 11], [22, 22, 22], [33], [77, 44]$
$[3, 3, 3, 3, 3, 3]$	1	$[3, 3, 3, 3, 3, 3]$

Curiosamente, esse algoritmo é capaz de encontrar o tamanho da maior subsequência crescente. Reflita sobre essa afirmação.

6. O Grande Sistema Planetário

O famoso astrônomo Typler precisa de sua ajuda para avançar em sua pesquisa sobre um sistema planetário distante. O problema é o seguinte: encontrar o menor número inteiro de anos, T , no qual a soma total de órbitas completadas por todos os n planetas do sistema seja maior ou igual a um valor k .

Dado um conjunto de períodos orbitais A_1, A_2, \dots, A_n (o tempo que cada planeta leva para completar uma volta) e um número alvo de voltas k , desenvolva um algoritmo que encontre o valor T . A complexidade esperada é de $O(n \cdot \log(M))$, onde M é a maior resposta possível.

Restrições:

- $1 \leq A_i \leq 10^9$ para todo $1 \leq i \leq n$
- $1 \leq k \leq 10^9$
- Consequentemente, $M \leq 10^{18}$

Entrada	Saída
$A = [3, 2, 6]$ e $k = 7$	8
$A = [3, 2, 6, 5, 8, 2, 1]$ e $k = 1234$	438

7. Otimização

Dado um vetor A com n inteiros, seu desafio é projetar um **algoritmo linear** que retorna o menor valor k que minimiza a soma:

$$\sum_{i=1}^n |A_i - k|$$

Restrições:

- $1 \leq n \leq 10^6$
- $|A_i| \leq 10^{18}$ para todo $1 \leq i \leq n$

Exemplos:

Entrada	Saída
$A = [3, 2, 6, 1]$	2
$A = [3, 2, 5, 8, 1, 3, 2, 3]$	3
$A = [33, 10]$	10