

Construção e Análise de Algoritmos

lista de exercícios 07

1. Contando intervalos invertidos

Imagine que nós temos uma coleção de intervalos $I_j = [x_j, y_j]$.

E imagine que os intervalos estão armazenados como uma lista de pares

I	<table><tr><td>(x_1, y_1)</td><td>(x_2, y_2)</td><td>(x_3, y_3)</td><td>(x_4, y_4)</td><td>(x_5, y_5)</td><td>(x_6, y_6)</td></tr></table>	(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	(x_4, y_4)	(x_5, y_5)	(x_6, y_6)
(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	(x_4, y_4)	(x_5, y_5)	(x_6, y_6)		

Nós dizemos que I_j e I_k são dois *intervalos invertidos* se

- $j < k$ e $y_k < x_j$



— (*i.e.*, o intervalo que aparece antes na lista está à direita do outro, sem interseção)

A tarefa consiste em

→ Contar o número total de pares de intervalos invertidos

Você consegue encontrar um algoritmo que resolve o problema em tempo menor que $O(n^2)$?

Você consegue encontrar um algoritmo que resolve o problema em tempo $O(n \log n)$?

2. Contando encontros

Imagine que existem n pessoas sobre uma linha



E imagine que a i -ésima pessoa se encontra na posição a_i e deseja ir para a posição b_i

— (*onde os números $a_1, \dots, a_n$ e $b_1, \dots, b_n$ são todos distintos*)

As pessoas começam a se mover simultaneamente, e andam todas com a mesma velocidade.

Um encontro acontece quando duas pessoas ocupam a mesma posição na linha — (*sendo que uma das pessoas pode estar parada nesse momento*)

A tarefa consiste em

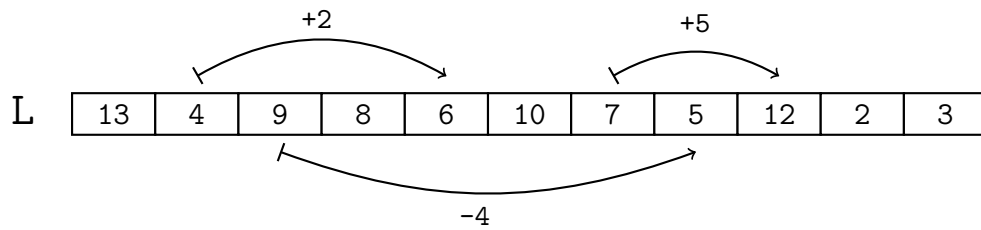
→ Contar o número total de pares de pessoas que se encontram

Você consegue um algoritmo que resolve esse problema em tempo menor que $O(n^2)$?

3. O maior salto ímpar

Imagine que L é uma lista de tamanho n que contém números inteiros.

E imagine que nós calculamos o salto de um elemento para outro assim



A tarefa consiste em

→ *Encontrar o par de elementos que dá o maior salto de tamanho ímpar*

Apresente um algoritmo que resolve esse problema em tempo menor que $O(n^2)$.

4. Variante do problema da faixa máxima

Imagine que L é uma lista que contém números positivos e negativos.

E imagine que k é um número qualquer.

A tarefa consiste em

→ *Encontrar a faixa de elementos consecutivos com soma $\leq k$
que tem a maior soma possível*

Apresente um algoritmo que resolve esse problema em tempo menor que $O(n^2)$.