

Construção e Análise de Algoritmos

lista de exercícios 01

Instruções:

2 ou 3 exercícios já bastam para praticar a construção e análise de algoritmos.

Não escolha exercícios fáceis demais, pois isso não ajuda muito.

Também não escolha exercícios difíceis demais.

E nem se preocupe com isso.

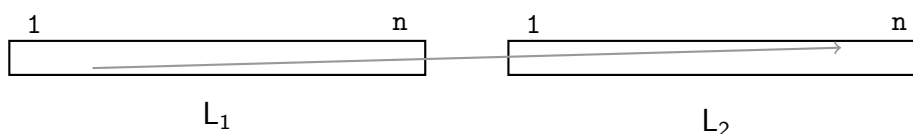
Porque aquilo que um dia é difícil, com o tempo se torna fácil

— (*e cada um sabe qual é o seu tempo ...*)

1. Ordenando duas listas

Imagine que uma quantidade muito grande $2n$ de elementos foi armazenada em duas L_1 e L_2 com tamanho n cada.

O problema consiste em reorganizar os elementos de modo que as duas listas, uma ao lado da outra, correspondam a uma única lista ordenada

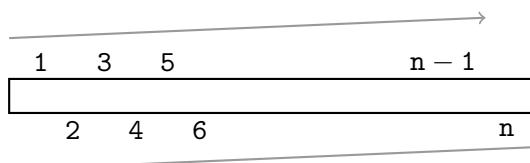


- Apresente um algoritmo que realiza essa tarefa sem utilizar uma lista auxiliar.
- Analise o tempo de execução do seu algoritmo.

2. Ordenação par-ímpar

A *ordenação par-ímpar* consiste em

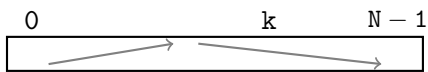
- ordenar os elementos das posições pares (em separado)
- ordenar os elementos das posições ímpares (em separado)



- Apresente um algoritmo que realiza essa tarefa sem utilizar uma lista auxiliar.
- Analise o tempo de execução do seu algoritmo.

3. Lista unimodal

A lista *unimodal* é uma lista onde os elementos primeiro crescem e depois decrescem — (ou *vice-versa*)

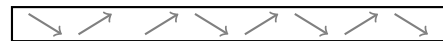
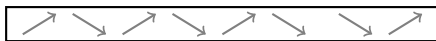


- Apresente um algoritmo que transforma a lista L em uma lista unimodal, onde a posição k é um parâmetro de entrada.
- Analise o tempo de execução do seu algoritmo.

4. Lista zigue-zague

A lista *zigue-zague* é uma lista onde não existem

- 3 elementos consecutivos em ordem crescente
- 3 elementos consecutivos em ordem decrescente



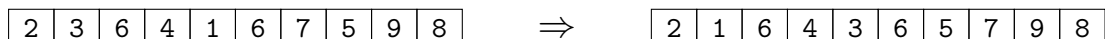
- Apresente um algoritmo que transforma a lista L em uma lista zigue-zague.
- Analise o tempo de execução do seu algoritmo.

5. Ordenação ímpar

A ordenação ímpar consiste em

- ordenar os números ímpares da lista
- mantendo os números pares nas suas respectivas posições

Por exemplo,

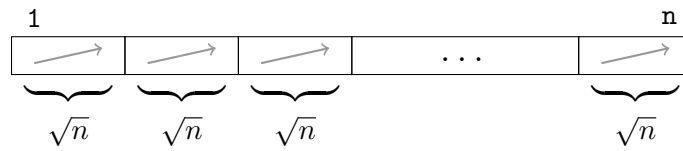


- Apresente um algoritmo que realiza essa tarefa sem usar uma lista auxiliar.
- Analise o tempo de execução do seu algoritmo.

6. Ordenação- $\sqrt{n}$

Imagine que L é uma lista de tamanho n , onde n é um quadrado perfeito.

O problema consiste em colocar a lista L na seguinte configuração



- Apresente um algoritmo que implementa a ordenação- $\sqrt{n}$.
- Analise o tempo de execução do seu algoritmo.