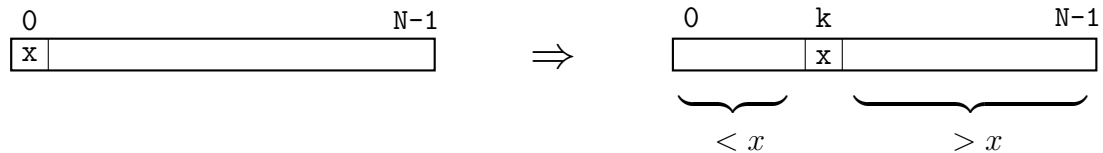


Construção e Análise de Algoritmos

Lista de exercícios 02

1. Ordenação via partição

Imagine que nós aplicamos o procedimento `Partição()` sobre a lista `L`



E imagine que depois nós ordenamos os dois lados da partição usando o algoritmo da bolha



Claramente, esse procedimento ordena a lista completamente.

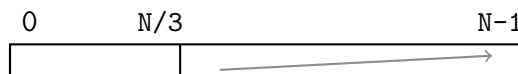
- Análise o tempo de execução desse algoritmo no melhor caso.
- Análise o tempo de execução desse algoritmo no pior caso.

2. Ordenação- $2n/3$

Imagine que nós ordenamos os primeiros $2n/3$ elementos da lista `L` usando o algoritmo da bolha



Daí, nós ordenamos os últimos $2n/3$ elementos da lista usando o algoritmo da bolha



Finalmente, nós ordenamos os primeiros $2n/3$ elementos da lista outra vez usando o algoritmo da bolha

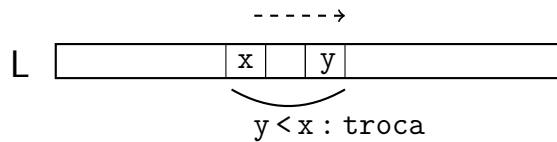


- Ao final desse processo, a lista está completamente ordenada?
Porque?
- Análise o tempo de execução desse procedimento.

3. Ordenação par-ímpar

Uma outra ideia para tornar o algoritmo da bolha mais eficiente consiste em acelerar o procedimento `Varredura()`.

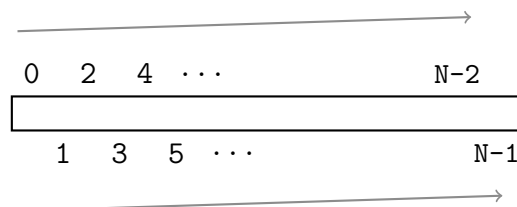
Quer dizer, a ideia consiste em comparar elementos à distância 2 — (*e trocá-los de posição, se necessário*)



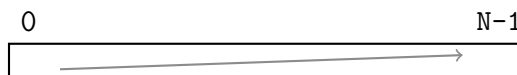
Na prática, isso corresponde a fazer a varredura primeiro sobre as posições pares, e depois sobre as posições ímpares.

E daí, se a gente executa o algoritmo da bolha com essa versão do procedimento `Varredura()`

- o algoritmo ordena as posições pares da lista em separado
- e ordena as posições ímpares da lista em separado



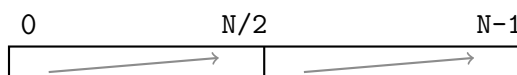
Finalmente, nós podemos aplicar a varredura tradicional repetidamente, até que a lista fique completamente ordenada



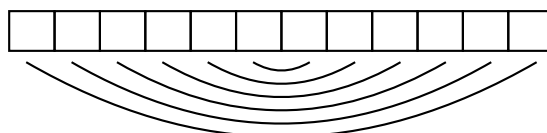
- Análise o tempo de execução desse algoritmo no melhor caso para a segunda etapa.
- Análise o tempo de execução desse algoritmo no pior caso para a segunda etapa.

4. Ordenação borboleta

Imagine que nós ordenamos as duas metades da lista L usando o algoritmo da bolha

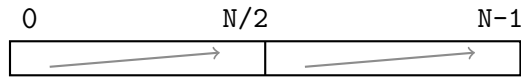


Daí, nós comparamos pares de elementos da seguinte maneira



sempre trocando os elementos de posição, quando o elemento da esquerda é maior do que o elemento da direita.

Finalmente, nós ordenamos as duas metades da lista outra vez usando o algoritmo da bolha



a) Ao final desse processo, a lista está completamente ordenada?

Porque?

b) Analise o tempo de execução desse procedimento.

5. Caixa mágica

Imagine que você tem uma caixa que ordena $\sqrt{n}$ elementos consecutivos de uma lista quase instantaneamente — (*i.e.*, em tempo $O(1)$)

a) Apresente um algoritmo que utiliza essa caixa para ordenar uma lista de tamanho n em tempo menor do que $O(n^2)$.

b) Analise o tempo de execução do seu algoritmo.

6. Ordenação paralela

Imagine que você tem um computador com muitos processadores.

E imagine que em um único passo de computação você pode comparar $n/2$ pares de elementos (distintos) e trocá-los de posição (se necessário).

a) Apresente um algoritmo que ordena uma lista de tamanho n nesse computador.

b) Analise o tempo de execução do seu algoritmo.