

Exercícios - Complexidade (Ingênua) de Algoritmos Iterativos

Análise de Algoritmos - DINF - UFPR

Exercício 1. Expresse (**informalmente**, como visto em sala) a função $n^3/1000 - 100n^2 - 100n + 3$ em termos da notação Θ .

Exercício 2. Considere ordenar n números armazenados num vetor A encontrando primeiramente o menor elemento de A e troque-o com o elemento em $A[1]$. Em seguida, encontre o segundo menor elemento de A , e troque-o com $A[2]$. Continue usando a mesma ideia para os primeiro $n - 1$ elementos de A .

- (a) Escreva um pseudo-código para este algoritmo (que é conhecido como **Selection Sort**).
- (b) Porque é preciso executar somente para os $n - 1$ primeiro elementos, ao invés de n ?
- (c) Dê o tempo de execução **exato** (assim como feito no InsertionSort em aula) de melhor-caso e pior-caso do **Selection Sort**.
- (d) Baseado no tempo de execução encontrado na item (c), expresse (**informalmente**, como visto em sala) o tempo na notação Θ .

Exercício 3. Considere o seguinte problema de busca:

Entrada: Uma seqüência de n números $A = (a_1, a_2, \dots, a_n)$ e um valor v .

Saída: Um índice i tal que $v = A[i]$ ou o valor especial NIL se v não aparece em A .

- (a) Escreva um pseudo-código para a busca linear, que varre a seqüência procurando por v .
- (b) Usando uma **invariante de laço**, prove que seu algoritmo é correto.
- (c) Quantos elementos da entrada devem ser verificados em média, assumindo que o elemento a ser buscado é igualmente provável de estar em qualquer posição do vetor?
- (d) E com relação ao pior-caso?
- (e) Qual é o tempo de execução da busca linear no caso médio e no pior caso usando (**informalmente**, como visto em sala) a notação Θ .