

Projeto e Análise de Algoritmos

José Ricardo Mello Viana

Aula 04
Análise de Algoritmos

2010.2

- 1 Análise de Algoritmos
- 2 Computador
- 3 Análise da ordenação por inserção
 - Tamanho da entrada
 - Tempo de execução
 - Exemplo
- 4 Análise do pior caso e do caso médio
- 5 Ordem de crescimento

- Prever os recursos que ele necessitará
- Memória, largura de banda, hardware
- Principal é o **tempo**
- Pela análise de vários podemos encontrar o mais eficiente
 - Inferiores são descartados
 - Mais de um viável pode ser encontrado

- Modelo de computação genérico
- Um único processador
- Sem execução concorrente
- Memória RAM comum (instruções uma após a outra)

- Tempo depende da entrada
- Geralmente cresce com o tamanho da entrada
- É tradicional descrever o tempo de execução como uma função do tamanho da entrada

- Mais natural: quantidade de itens
- Multiplicação de inteiros: quantidade de bits
- Às vezes com 2 números ao invés de 1: se for um grafo, número de arestas e vértices

- Número de operações primitivas ou “etapas” executadas
- Devemos definir de forma que seja independente da máquina
- Período *constante* para cada linha
- Estruturas de repetição: custo *linear*

insertion_sort(A):

for $j \leftarrow 2$ to $\text{comprimento}[A]$

do $\text{chave} \leftarrow A[j]$

//Inserir $A[j]$ na sequência ordenada $A[1..j - 1]$

$i \leftarrow j - 1$

while $i > 0$ e $A[i] > \text{chave}$

do $A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

$A[i + 1] \leftarrow \text{chave}$

- Melhor caso: vetor já ordenado (custo linear)
- Pior caso: vetor em ordem inversa (custo quadrático)
- Nos concentraremos apenas no **no tempo do pior caso**
- Tempo de execução mais longo para qualquer entrada de tamanho **n**
 - Limite superior
 - Para alguns algoritmos ocorre com bastante frequência
 - Muitas vezes o “caso médio” pode ser tão ruim quanto o pior caso

- Em alguns casos particulares estudaremos o **caso médio**
- Difícil prever o que constitui uma entrada “média”
- Entradas serão igualmente prováveis
- **Algoritmos aleatórios**

- Instruções simples: custo constante
- Consideraremos apenas o termo de mais alta ordem
 - Para $an^2 + b^n + c$, apenas an^2
- Também ignoramos a constante: n^2
- Dizemos que o `insertion_sort` possui pior caso $\theta(n^2)$
 - theta de n ao quadrado
- Dizemos que um algoritmo é “melhor” que outro quando tem ordem de crescimento mais baixa