

Período: 2006.1

25 de abril de 2006

Horário: 2as-feiras e 4as-feiras de 19 às 21 horas

## ANÁLISE DE ALGORITMOS (INF 1309)

### 1º Trabalho de Implementação (T1)

**Data de Entrega: 19 de maio**

#### Descrição

Este trabalho prático consiste em desenvolver para os algoritmos para a resolução de problemas sobre conjuntos e vetores contendo números inteiros positivos.

1. Sejam  $S_1$  e  $S_2$  conjuntos cujos elementos são números inteiros e onde  $|S_1| + |S_2| = n$ . Dado um inteiro  $x$ , propor um algoritmo para determinar se existem  $e_1 \in S_1$  e  $e_2 \in S_2$  tais que  $e_1 + e_2 = x$ .
  - (a) Proponha e implemente um algoritmo de complexidade  $\Theta(n^2)$  e apresente a análise que permite concluir esta complexidade.
  - (b) Proponha e implemente um algoritmo de complexidade  $O(n \log n)$  e apresente a análise que permite concluir esta complexidade.
  - (c) Suponha agora que  $S_1$  e  $S_2$  estejam ordenados. Proponha e implemente um algoritmo de complexidade  $O(n)$  e apresente a análise que permite concluir esta complexidade.
  - (d) Execute os algoritmos implementados sobre conjuntos  $S_1$  e  $S_2$  de tamanho total 1000, 2000, 5000, 10000, 20000, 50000 e 10000. Apresente os tempos de CPU para cada execução de cada algoritmo, organize numa tabela.
  - (e) Analise os resultados experimentais obtidos e compare com as complexidades teóricas.
2. Sejam  $S_1$  e  $S_2$  conjuntos cujos elementos são números inteiros e onde  $|S_1| = |S_2| = n$ .
  - (a) Apresente e implemente um algoritmo de complexidade  $\Theta(n^2)$  para determinar a união de  $S_1$  e  $S_2$ . Nenhum elemento deve aparecer mais de uma vez.
  - (b) Apresente e implemente um algoritmo que execute em  $O(n \log n)$ .
  - (c) Execute os algoritmos implementados sobre conjuntos  $S_1$  e  $S_2$  de tamanho 1000, 2000, 5000, 10000, 20000, 50000 e 10000. Apresente os tempos de CPU para cada execução.
  - (d) Analise os resultados experimentais obtidos e compare com as complexidades teóricas.
3. Busca em Vetores Ordenados
  - (a) Escreva um algoritmo de "busca ternária" que primeiro testa se o elemento da posição  $n/3$  é igual ao elemento procurado  $x$  e então, se necessário, testa o elemento da posição  $2n/3$  para verificar a igualdade com o elemento  $x$  e, possivelmente, reduzir o tamanho do conjunto para um terço do original.

- (b) Implemente um algoritmo de Busca Binária.
- (c) Execute os algoritmos implementados sobre vetores de tamanho 1000, 2000, 5000, 10000, 20000, 50000 e 10000. Apresente os tempos de CPU para cada execução. Apresente também o número de comparações e recursões realizadas em cada execução.
- (d) Analise os resultados experimentais obtidos e compare com as complexidades teóricas.

Os algoritmos acima devem ser desenvolvidos e seu desempenho experimental deve ser analisado com respeito ao tempo de CPU. O desenvolvimento destes códigos e a análise devem seguir os seguintes roteiros:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Explicar a fundamentação do algoritmo e justificar a sua corretude. Apresentar e explicar a complexidade teórica esperada para cada algoritmo.
- Apresentar gráficos com os tempos de CPU para a execução de cada um dos casos teste (observe que tempos de CPU inferiores à 5s não são confiáveis, nesse caso repita a execução do teste até que o tempo ultrapasse 5s, divida então o tempo de CPU obtido pelo número de execuções que foram necessárias);
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.

A corretude código será testada sobre instâncias cujo gerador será fornecido e poderá ser encontrado na página do curso. Também estará disponível na página do curso uma classe para a obtenção do tempo de CPU gasto.

O trabalho entregue deve conter:

- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos acima, comentários e análises sobre a implementação e os testes realizados (papel).
- A impressão dos códigos fonte (papel).
- Um e-mail contendo os códigos fonte, os executáveis e o documento em versão eletrônica, correspondentes (o e-mail deve ser enviado para [marcus.poggi@gmail.com](mailto:marcus.poggi@gmail.com) e é obrigatório o uso do ASSUNTO (ou SUBJECT) AA061T1).
- O trabalho pode ser feito em grupos de até 2 alunos.