



## **Laboratório 1 - Conectividade**

Semana de 21 a 25 de Setembro de 2015

Duração: 2 horas

Na primeira parte deste laboratório propõe-se a execução de um programa que implementa um conjunto de algoritmos simples, estudados nas aulas teóricas, sobre o Problema da Conectividade. Pretende-se que os alunos façam uma análise cuidadosa destes programas, a nível semântico e sintático. Através desta análise, deverão ser capazes de identificar o que cada algoritmo faz, introduzindo código pertinente para obter a informação sobre o funcionamento do programa em termos de complexidade. Deve ser assumido que a complexidade dos algoritmos é função do número de operações elementares associadas aos procedimentos de procura e de união, considerando-se como operações elementares apenas os acessos aos elementos das tabelas de dados.

Na segunda parte do laboratório pretende-se que os alunos, fazendo uso do conhecimento dos algoritmos estudados na primeira parte do laboratório, desenvolvam código para obter informações adicionais respeitantes ao problema da conectividade.

Os dados são inicialmente lidos de um ficheiro de entrada. A sintaxe deste ficheiro é muito simples: na primeira linha do ficheiro é indicado o número de objectos/nós do problema. Nas linhas seguintes são indicados os pares de ligações. O programa deve ler todo o ficheiro até esgotar os dados.

1. O ficheiro `class2.c` abre e lê o ficheiro de entrada fornecido como argumento na linha de comando. O programa contém um menu que permite a escolha entre várias opções de algoritmos para efectuar o teste de conectividade. Analise o fluxograma de um algoritmo genérico para o problema da conectividade que se apresenta na Fig. 1. Para o caso particular do algoritmo de Procura Rápida - QuickFind (QF) analise ainda os fluxogramas dos procedimentos de Procura e de União. O código deste e dos restantes algoritmos encontra-se no ficheiro `connectivity.c`.
2. Pretende-se analisar o desempenho dos diversos algoritmos num conjunto de problemas cujos pares de ligações se encontram nos ficheiros de dados fornecidos. Adicione código para contabilizar o número de operações elementares (leituras e escritas das tabelas de dados) associados aos procedimentos de procura e de união efectuadas. Nos algoritmos de União Rápida Equilibrada - Weighted Quick Union (WQU) e União Rápida com Compressão de Caminhos - Compressed and Weighted Quick Union (CWQU) contabilize as operações associadas aos procedimentos de equilíbrio e de compressão como operações de união. No final da execução do programa, deve escrever estes valores para o terminal (`stdout`). Apresente os resultados obtidos completando a Tabela 1, indicando para cada ficheiro e algoritmo os valores contabilizados de nós, pares de entrada, ligações e operações de procura e união.
3. Verifique se os resultados obtidos experimentalmente estão de acordo com os resultados teóricos. Para isso, introduza os dados da Tabela 1 numa folha de cálculo e utilize os instrumentos de regressão nela disponibilizados.
4. Pretende-se agora que no final do programa se identifiquem os diferentes conjuntos resultantes da união usando o QuickFind. Para tal, desenvolva código para que o programa escreva no terminal (`stdout`) os elementos de cada conjunto. Os conjuntos deverão ser impressos em linhas distintas, sendo os elementos de cada conjunto separados por um hífen. Finalmente, deverá ser impresso o número de conjuntos.

Exemplo com 3 conjuntos distintos:

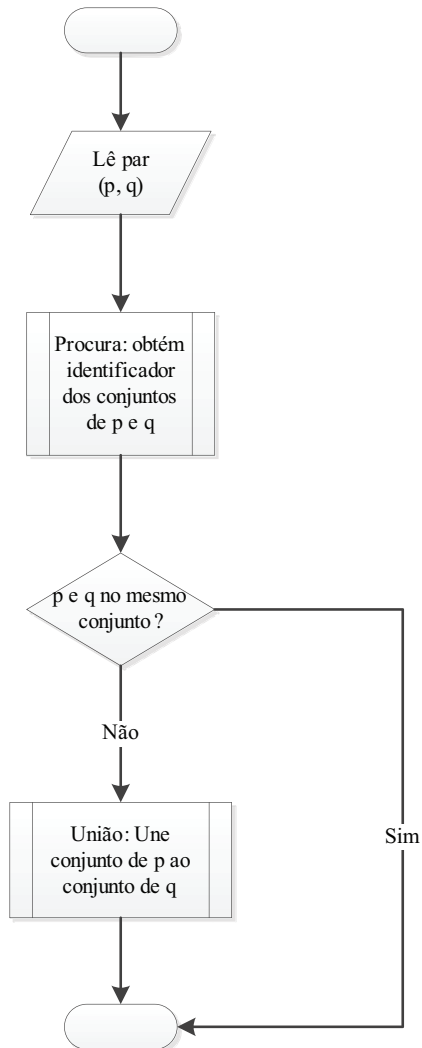
0-1-5-6-7-8-9

2-4

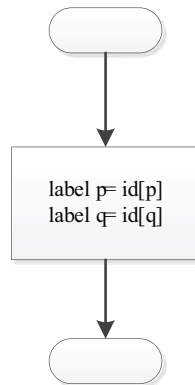
3

Número de conjuntos: 3

Algoritmo genérico  
para o problema da conectividade



QuickFind: Procura



QuickFind: União

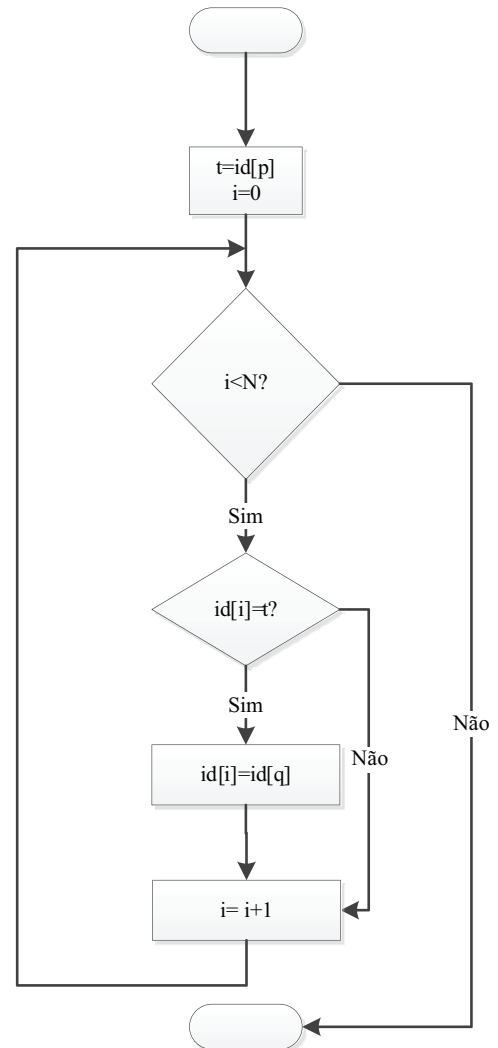


Figura 1: Fluxogramas

Ficheiro	Nós	Pares	Ligações	Quick Find		Quick Union		WQU		CWQU	
				Find	Union	Find	Union	Find	Union	Find	Union
10.txt											
100.txt											
1000.txt											
10000.txt											
a.txt											
b.txt											
c.txt											
d.txt											
e.txt											

Tabela 1: Resultados experimentais