



## Laboratório 1 - Conectividade

Semana de 23 de Setembro de 2019

Duração: 2 horas

Este laboratório foca-se, de forma prática e experimental, nos algoritmos estudados nas aulas teóricas sobre o Problema da Conectividade. É fornecido o código para executar quatro algoritmos ou variantes, bem como para ler ficheiros de dados, dos quais também é fornecido um conjunto de teste.

Os objectivos, detalhados mais abaixo, são analisar os algoritmos e obter dados experimentais da sua complexidade de execução, e acrescentar apresentação de resultados num dos casos. Será necessário modificar o código fornecido e correr o programa sobre dados de teste.

O ficheiro `labconn.c` abre e lê o ficheiro de entrada fornecido como argumento na linha de comando. O programa apresenta no terminal um menu para escolha de algoritmo para resolver o problema de conectividade.

A sintaxe do ficheiro de entrada 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.

O ficheiro `connectivity.c` tem código para quatro algoritmos:

- Procura Rápida - *QuickFind* (QF)
- União Rápida - *Quick Union* (QU)
- União Rápida Equilibrada - *Weighted Quick Union* (WQU)
- União Rápida com Compressão de Caminhos - *Compressed and Weighted Quick Union* (CWQU)

**Nota:** Ao analisar o desempenho dos diversos algoritmos, para simplificar vamos assumir que uma operação elementar corresponde a um acesso (leitura ou escrita) a uma posição da tabela de dados.

O trabalho a efectuar no laboratório consiste nos quatro itens seguintes:

1. 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 (QF) analise ainda os fluxogramas dos procedimentos de Procura e de União.

Correlacione com o código fornecido, identificando com o que cada algoritmo faz, e calcule o que será expectável do programa em termos de complexidade.

2. Adicione código para contabilizar o número de operações elementares nos procedimentos abstractos de procura e de união efectuadas. Obtenha igualmente o total de operações elementares de cada algoritmo.

Nos algoritmos WQU e CWQU contabilize também as operações associadas aos procedimentos de equilíbrio e de compressão como operações de união.

No final do programa, estes valores devem ser escritos para o terminal (*stdout*). Pretende-se obter dados suficientes para preencher a Tabela 1 ou similar, indicando para cada ficheiro e algoritmo os valores contabilizados de nós, pares de entrada, ligações e operações de procura e união.

**Nota:** A execução dos algoritmos de Procura Rápida e União Rápida pode ser muito demorada nos ficheiros de maiores dimensões pelo que deverá ser feita antes do laboratório.

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. Sugere-se em particular fazer uma visualização com *scatterplot* e obter uma *trendline* para os dados, mostrando a equação e o factor de adaptação.

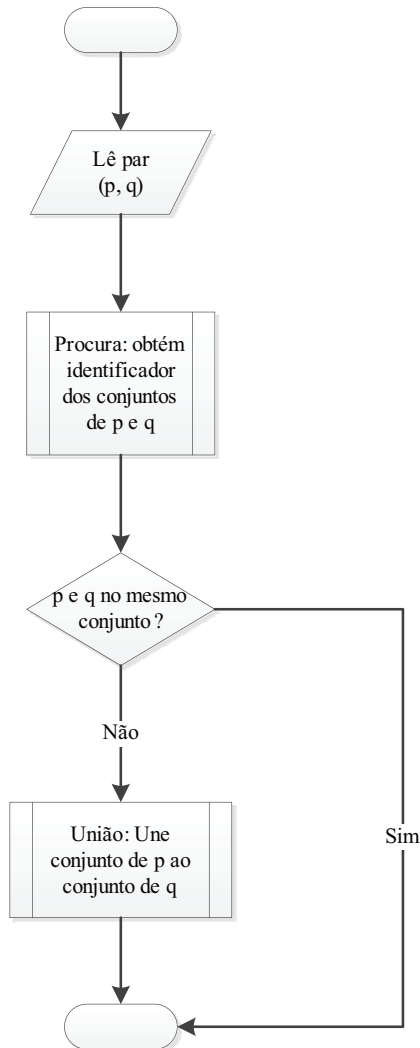
4. Pretende-se agora que no final sejam apresentados os diferentes conjuntos resultantes da união usando o QuickFind. Para tal, desenvolva código para que o programa escreva no terminal (*stdout*) a solução obtida.

Cada conjunto deverá ser impressos numa linha distinta, sendo os elementos separados por um hífen. Para terminar deverá ser impresso o número de conjuntos.

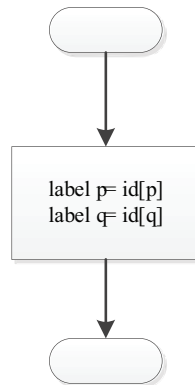
Exemplo com 3 conjuntos distintos:

```
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

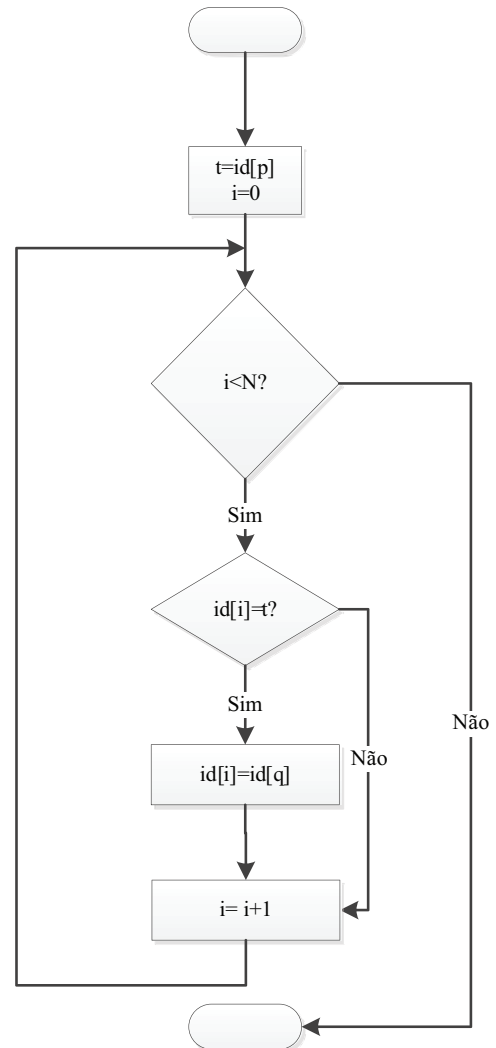


Figura 1: Fluxogramas

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

Tabela 1: Resultados experimentais