

Exercício 27 de Grafos

Rafael Crivellari Saliba Schouery NUSP 538223

13 de maio de 2007

1 Programa

O programa está em anexo no pacote deste arquivo. Ele consiste de um arquivo Makefile que permite gerar os executáveis *tarjan* e *kosaraju* através de **make tarjan**, **make kosaraju** ou **make all**.

Os arquivos fonte estão divididos em **ex27.c**, responsável pela análise dos parâmetros passados pela linha de comando e pela chamada das funções necessárias para ler o grafo e descobrir os componentes fortemente conexos, **GRAPH.c** e **GRAPH.h** responsáveis pela implementação da **ADT** de grafo como lista de adjacência, **graph-common.c** e **graph-common.h** responsáveis pelas funções de grafo que não dependem da implementação e finalmente **strong-comp.h**, **tarjan.c** e **kosaraju.c** responsáveis pela descoberta dos componentes fortemente conexos.

2 Teste Empírico

Foram executados, para cada valor de m sugerido ($0,9n$; $0,95n$; n ; $1,05n$; $1,1n$), 100 testes com $n=100.000$ para cada um dos algoritmos utilizando o `g_random` com a opção `-d`. As médias de duração da execução de cada teste podem ser vistas na figura 1. Podemos ver que o algoritmo de Kosaraju é mais lento do que o algoritmo de Tarjan. Isso se deve ao fato de que o algoritmo de Kosaraju faz duas buscas em profundidade e uma inversão no grafo, enquanto que o algoritmo de Tarjan faz apenas uma busca e olha para cada vértice duas vezes (uma quando empilha, outra quando desempilha). Em média o algoritmo de Kosaraju se mostrou 128% mais lento que o algoritmo de Tarjan.

Abaixo está a tabela gerada pelo experimento e em anexo no pacote deste arquivo está o arquivo `exec.rb` que gerou as chamadas para criar os dados. Este arquivo foi feito em Ruby pela praticidade da linguagem.

m	Tarjan	Kosaraju	Porcetagem (K/T)
0,9	0,1446 s	0,1828 s	126%
0,95	0,1426 s	0,181 s	127%
1	0,1622 s	0,2082 s	128%
1,05	0,237 s	0,3155 s	133%
1,1	0,2252 s	0,2833 s	126%

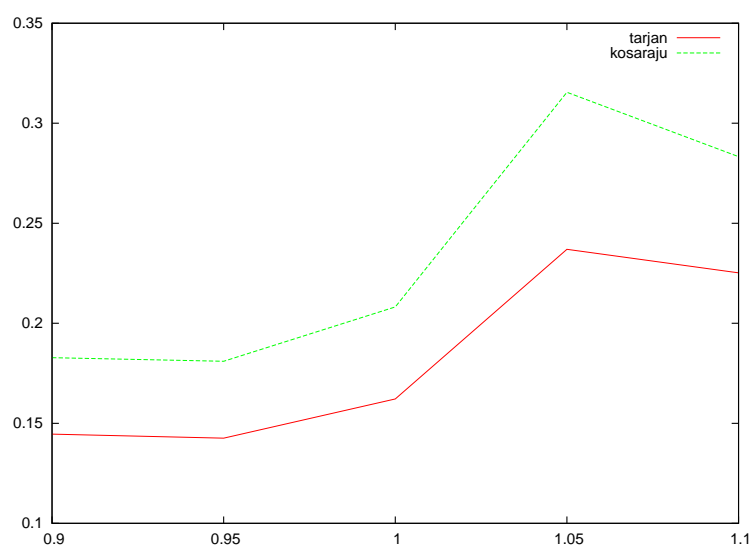


Figura 1: Média de duração dos testes