# MAC0438 - Programação Concorrente - 1s2012
# Relatório

Thiago de Gouveia Nunes
Wilson Kazuo Mizutani

May 13, 2012

## Contents

# 1 Introdução

Esse relatório trata das decisões tomadas na implementação desse EP. Também fornece uma saída do programa para a entrada de exemplo fornecida no enunciado, além de explicitar a localização no código da implementação da barreira simétrica usada. Informações sobre como como compilar, as dependÊncias necessárias e o modo de uso do programa encontram-se no arquivo LEIAME.

# 2 Solução desenvolvida

## 2.1 Ideia geral

A ideia geral da nossa solução se divide em duas partes:

1. Estender o algoritmo de busca em largura (Breadth-First Search, um caso particular do algoritmo de Dijkstra no qual todas as arestas possuem custo 1) para que ele encontre não só o menor caminhos, mas sim os **n** menores caminhos.

2. Refatorar esse algoritmo para usar programação concorrente, de tal maneira que cada thread seja responsável por tentar encontrar um novo caminho e depois sincronizar com as demais, criando assim um processo iterativo.

## 2.2 Estendendo a busca em largura

Basicamente, aproveitamos a propriedade da busca em largura na qual um novo vértice retirado da fila está sendo visitado pela primeira vez e através do menor caminho.

Estendemos o algoritmo para ter uma fila de caminhos ao invés de vértices, por conveniência, e ao invés de deixarmos de visitar um vértice após passar po ele apenas uma vez, o fazemos após **n** vezes (o que significa que ele já tem n caminhos mínimos terminando nele).

Assim, a propriedade do nosso algoritmo (ainda no caso não-concorrente) seria que um novo caminho retirado da fila é o próximo menor caminho que termina no mesmo último vértice dele. Mas como o enunciado do EP pedia que cada thread cuidasse de apenas um caminho, mudamos isso de forma que, na verdade, insere-se apenas candidatos a caminho na fila. A propriedade fica portanto que um novo candidato retirado da fila *p*ode ser o próximo menor caminho que termina no seu último vértice, *c*ontanto que ele não seja um ciclo.

Uma consequência disso é que o programa vai com certeza ficar menos eficiente, pois há menos restrições sobre quem entra na fila, e portanto ela potencialmente terá mais elementos do que na maneira anterior.