



Relatório

Projeto 1

Análise e Síntese de Algoritmos

Miguel Lourenço – 72588
Henrique Lourenço - 77459

Introdução

O objectivo deste projeto é, dado um conjunto de pessoas e relações de colaboração, encontrar (i) o maior valor de número de Edrões e (ii) uma sequência de linhas correspondente ao número de pessoas com o mesmo número de Edrões.

Estruturas utilizadas

Utilizámos um array de vértices, chamado “Graph”, onde cada “vertex” tem armazenada a cor, distância e uma fila de arcos.

Esta Fila tem o nó inicial e final e cada nó tem um ponteiro para o vértice e para o próximo nó.

Por último utilizámos uma fila semelhante, chamada “Priority Queue”, que serve para saber a fila dos vértices sobre os quais ainda vamos iterar o algoritmo.

Algoritmo principal (BFS)

O algoritmo BFS (Breadth-first search)

Este algoritmo explora sistematicamente os vértices de forma a descobrir todos os vértices atingíveis desde o vértice inicial.

A distância a cada vértice é calculada pelo menor número de arcos entre o vértice inicial e o vértice atingível.

Descrição do processo:

Inicialização:

- Começa por ler o número de vértices, arestas e o vértice inicial, do input.
- Cria um vector com todos os vértices
- Lê todos os arcos do input e cria os arcos entre os vértices, na fila.
- Por fim cria uma “Priority queue” e insere o vértice inicial nesta fila.

Ciclo principal:

- O algoritmo começa com a estrutura dos nós vizinhos já criada, ou seja, o ficheiro já foi totalmente lido.
- Lê o primeiro vértice da “Priority Queue” e retira-o da fila.
- Lê cada vértice adjacente e caso este vértice esteja a branco, passa-o a cinzento e incrementa a distância em um valor.
- Insere o vértice adjacente no fim da “Priority Queue”.
- Após de ler todos os vértices adjacentes, passa a cor do vértice que se estava a trabalhar para preto.

Impressão:

- Imprime a maior distância ao vértice inicial na primeira linha, nas seguintes imprime uma sequência de linhas onde a linha corresponde ao número de Erdős com distância ao vértice inicial igual ao número da linha.

Avaliação experimental:

Esta solução passa com sucesso aos 16 testes presentes no sistema Mooshak. Abaixo encontra-se uma tabela com o tempo de execução do programa (em milissegundos) com inputs com diferentes V (vértices) e A (arcos).

V	A	10000	25000	50000	100000	250000
10000		4.0	8.0	-	-	-
25000		-	12.0	40.0	44.0	-
50000		-	-	48.0	144.0	
100000		-	-	-	56.0	136.0
250000		-	-	-	-	172.0

Os valores acima apresentados são o resultado da execução do comando Unix time e correspondem à soma dos valores user com sys. Verifica-se que o algoritmo é linear no tamanho do input. O programa fica mais lento com o aumento dos arcos do que com o aumento dos vértices, devido à natureza do algoritmo BFS.