Faculdade de Engenharia da Universidade do Porto

Tema 6:

Supermercado ao Domicílio

Concepção e Análise de Algoritmos

2016/2017 -- Mestrado Integrado em Engenharia Informática e Computação:

Turma 5, Grupo G:

Afonso Bernardino da Silva Pinto (up201503316@fe.up.pt)

Daniel Pereira da Silva (up201503212@fe.up.pt)

Tomás Sousa Oliveira (up201504746@fe.up.pt)

Docentes:

Ana Paula Lopes arocha@fe.up.pt Rosaldo Rossetti rossetti@fe.up.pt

Rui Camacho rcamacho@fe.up.pt

Descrição do Tema

Parte 1:

Uma rede de supermercados deseja inovar e passa a permitir que os seus clientes realizem as suas compras de supermercado pela Internet, sendo depois as mesmas entregues no domicílio do cliente. Para rentabilizar o serviço, o supermercado tenta agrupar o maior número de compras numa única viagem, distribuindo as compras para os clientes nas suas respectivas moradas.

Elabore um sistema que permite ao supermercado avaliar diferentes estratégias de entregas ao domicílio. Os camiões poderão todos partir de um único supermercado, ou poderá haver vários supermercados a manterem o seu próprio veículo de entrega, procurando-se agrupar os clientes mais próximos de uma dada sucursal.

Avalie a conectividade do grafo, a fim de evitar que clientes sejam atendidos por supermercados para os quais não haverá itinerário possível. Para além de tentar-se minimizar o itinerário de distribuição das compras, pretende-se também realizar o maior número de entregas numa única viagem, com a condição de terem todas as compras de ser entregues no mesmo dia em que foram realizadas.

Considere a possibilidade de utilizar mapas reais, extraídos do OpenStreetMaps (www.openstreetmap.org), localizando, no mapa, as sucursais da rede de supermercados e as moradas dos respectivos clientes.

Dados:

Construção de um grafo, G = (T,E), em que:

Vértices (T): corresponde aos lugares no mapa;

Arestas (E): corresponde às ruas entre as localizações.

Parte 2:

Para a segunda parte deste trabalho, considere que as ruas têm nomes, por exemplo "Rua de Dr Roberto Frias" ou "A1", e que pertencem a um dado distrito, por exemplo, "Porto"; os supermercados também terão um nome, ou pertencerão a uma cadeia (e.g. Pingo Doce, Continente, etc.). Considerando que os supermercados estão localizados em esquinas de cruzamentos entre ruas, estenda o trabalho realizado com funcionalidades apropriadas que permitem a consulta da existência ou não de um supermercado num determinado cruzamento entre ruas. Implemente esta funcionalidade, considerando tanto pesquisa exata, assim como pesquisa aproximada, das strings identificativas dos nomes das ruas fornecidas, e dos supermercados. Para pesquisa exata, caso os nomes das ruas (ou dos supermercados) não existam, deverá retornar mensagem de lugar desconhecido. Para a pesquisa aproximada, deverá retornar os nomes de ruas mais próximos, ordenados por similaridade, onde poderá haver um supermercado.

Estas novas funcionalidades deverão ser integradas no trabalho já realizado para a primeira parte. Avalie a complexidade (teórica e empiricamente) dos algoritmos implementados em função dos dados de input usados..

Por outras palavras, o principal objetivo do projeto é introduzir uma pesquisa dinâmica que facilite a procura de supermercados nas redondezas do utilizador. Este deverá assim poder pesquisar de forma exata por determinado par de ruas ou realizar uma pesquisa aproximada, onde qualquer possível erro na introdução do nome das ruas, passará a despoletar um conjunto de 'ruas-sugestão' em detrimento da indicação de não encontrado.

Introdução

Este projeto foi-nos proposto no âmbito da disciplina Concepção e Análise de Algoritmos, leccionada no 2º ano do Mestrado Integrado em Engenharia Informática e Computação. À criação de um supermercado ao domicílio é agora adicionado a capacidade de pesquisa em strings (exata e aproximada).

Para procedermos à expansão do programa decidimos fazer umas pequenas reformulações às estruturas de dados utilizadas no fase transata, bem como, a introdução de um algoritmo de pesquisa aproximada de strings, *Edit Distance*.

Palavras-Chave

Supermercado; Pesquisa; String; Grafo; Eclipse; C++; Concepção e Análise de Algoritmos; FEUP; Domicílio; GraphViewer; Edit Distance; Exata; Aproximada;

Agradecimentos

Este projeto foi resultado de diversas contribuições e colaborações, dada de forma direta e indireta, mas todas elas essenciais à sua realização. Gostaríamos assim de expressar os nossos sinceros agradecimentos a todos os que tornaram possível este trabalho, especialmente à professora Ana Paula Rocha, pela orientação dada e valioso acompanhamento constante durante o desenvolvimento do projeto; e também aos docentes de forma geral pela disponibilização do *OSM2TXT Parser* e do *GraphViewer*, ferramentas que facilitaram grandemente o desenvolvimento deste projeto.

Índice

1. Formalização do Problema

1.1. Dados de Entrada

A entrada de dados é feita por intermédio de ficheiros, a saber:

- nodes.txt
  - Este ficheiro faz parte do mapa. Nele, são guardados os nós.
- roads.txt
  - Este ficheiro faz parte do mapa. Nele, são guardados os nomes das ruas.
- geoms.txt
  - Este ficheiro faz parte do mapa. Nele, são guardadas as arestas.
- clients.txt
  - Neste ficheiro é guardada a localização de cada cliente.
- supermarkets.txt
  - Neste ficheiro é guardada a localização de cada supermercado.

1.2. Restrições

A localização dos supermercados foi forçosamente alterada para coincidir com cruzamentos entre ruas de forma a implementar a funcionalidade de pesquisa conforme requisitado.

1.3. Função Objetivo

Encontrar por pesquisa exata ou aproximada o supermercado pesquisado quer por via direta (introdução do nome do supermercado) quer por via indireta(introdução das ruas que se cruzam junto ao supermercado).

1.4. Resultados Esperados

Espera-se que o programa seja capaz de identificar os supermercados pesquisados por qualquer uma das vias supracitadas, e também que forneça sugestões adequadas aquando de pesquisas falhadas.

2. Descrição da Solução Implementada

O principal objetivo deste projeto era construir um mecanismo robusto e eficaz para a pesquisa de um supermercados num mapa de uma dada região. Para isso foi implementado o algoritmo *Edit Distance* melhorado: um algoritmo de pesquisa aproximada de strings (algoritmo implementado na classe SuperMarketChain).

2.1. Análise do algoritmo *Edit Distance* (melhorado)

O algoritmo *Edit Distance* recebe como argumentos duas strings:

- Pattern - a palavra a procurar;
- Text - o texto onde procurar a palavra;

O algoritmo retorna um inteiro:

- d[n] - a distância entre as strings;

O algoritmo recorre a um array de tamanho |T| e a duas variáveis auxiliares para guardar os valores relevantes para o preenchimento desse array.

Como o algoritmo analisa todos os caracteres, tanto P como de T, a complexidade temporal deste é: O(|P|.|T|).

A complexidade espacial é de O(|T|), uma vez que apenas é utilizado um array.

2.2. Análise do algoritmo pesquisa exata

Quando o utilizador opta, no menu, por fazer uma pesquisa exata debruça-se com 2 opções, pesquisa por nome do supermercado, ou por nome das ruas envolventes.

No caso da pesquisa por supermercado, o mecanismo responsável por dar resposta percorre um vetor com informação de todos os supermercados (previamente preenchido na fase de 'Loading Resources' do programa) e compara de forma 'grosseira' o nome do supermercado com o nome fornecido. Complexidade Temporal: O(|N|), Complexidade Espacial: O(|N|).

Já no caso da pesquisa por nome das ruas, o processo recorre ao grafo realizado na fase 1 do projeto e avalia de todos os vértices existentes as ruas que neles começam até encontrar o nome de uma delas, aí se algum dos destinos possíveis for um supermercado, percorre todas as ruas que envolvem esse ponto em busca de uma com o mesmo nome que a 'road2', nome da segunda rua indicada. Complexidade Temporal: O(|N|.|M|.|P|), Complexidade Espacial: O(|N|.|M|.|P|).

2.3. Análise do algoritmo pesquisa aproximada

Tal como no caso anterior, quando o utilizador opta, no menu, por fazer uma pesquisa aproximada debruça-se com 2 opções, pesquisa por nome do supermercado, ou por nome das ruas envolventes.

O mecanismo adotado é idêntico para qualquer das situações. Cada string introduzida é avaliada pela função getApprString, que é responsável por criar um vetor de *scores* que posteriormente ao seu preenchimento é ordenado crescentemente conforme a distância calculada através do já mencionado algoritmo Edit Distance.

Feita esta avaliação são mostradas as sugestões ao utilizador e procede-se ao cálculo exato tendo em conta a resposta do utilizador.

Complexidade Temporal: O(|N| . log(|N|)) + O(Pesquisa Exata).

Complexidade Espacial: O(|N|).

3. Diagrama de Classes

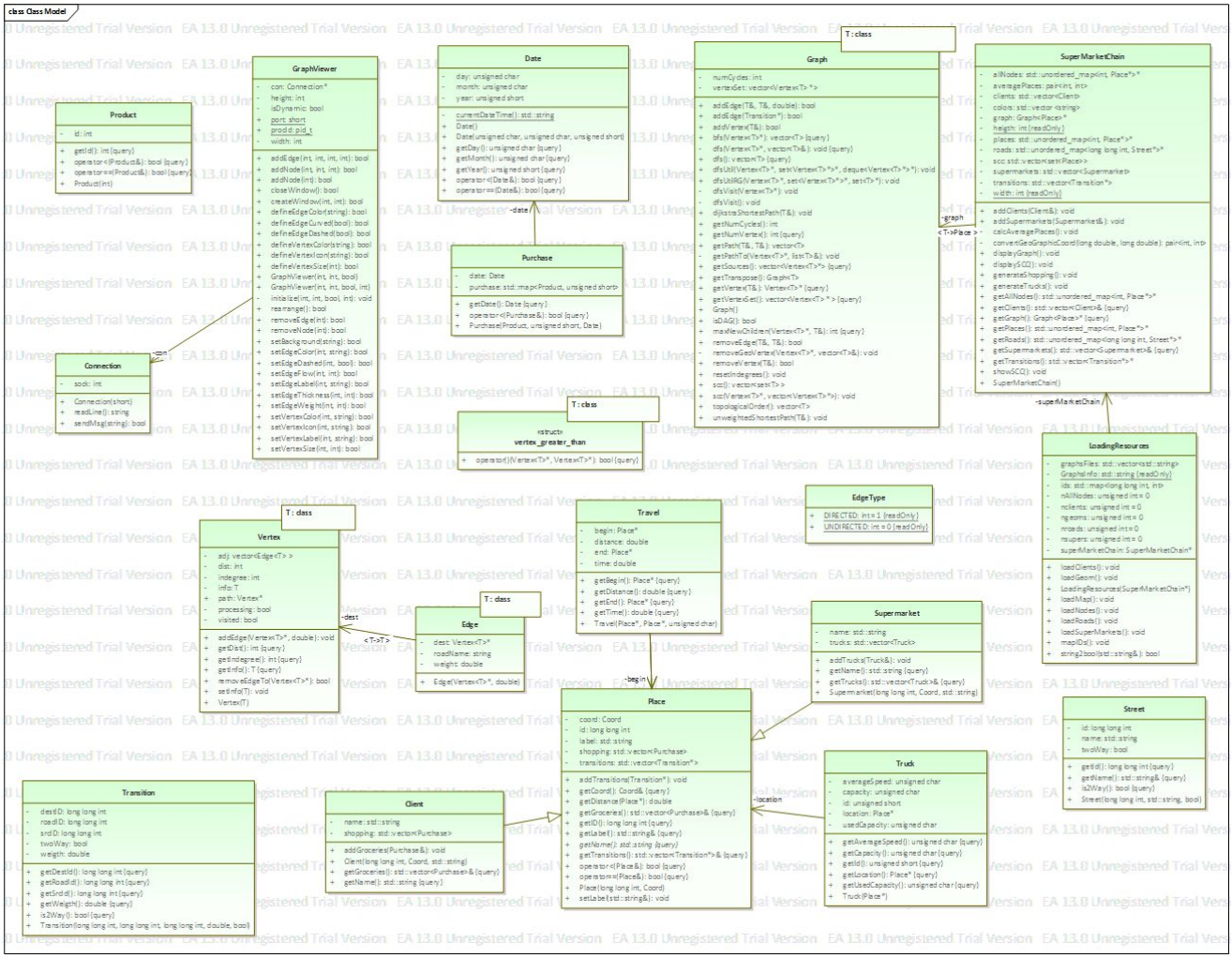

Imagem com melhor resolução em: http://imgur.com/KA05f75

4. Casos de Utilização identificados para a aplicação

- Mostrar Grafos com a localização dos Clientes e Supermercados no mapa
- Gerar Compras
- Ver a lista de Clientes
- Ver a lista de Supermercados
- Mostrar a rota dos Camiões até aos Clientes
- Visualizar uma informação detalhada da rota
- Visualizar as componentes fortemente conexas

Não presentes na versão anterior:

- Procurar por nomes de ruas com nome exato
- Procurar por nomes de ruas com nome aproximado

5. Principais Dificuldades

Para a segunda parte do trabalho, a dificuldade diminuiu consideravelmente. O nosso grupo vê como principal razão para essa queda a existência de um conjunto prévio de estruturas de dados bem preenchidas e objetivas.

O principal contratempo com que nos deparamos era alheio ao nosso trabalho uma vez estava relacionado com o GraphViewer que revelou alguns problemas no posicionamento no nome das ruas.

6. Esforço de cada Elemento

Afonso Pinto (33.33%): 'Bug Wrangler' - Solucionou a maior parte dos problemas que surgiram na implementação dos algoritmos; Responsável pela elaboração do relatório; Responsável pela implementação de pequenas partes dos algoritmos de pesquisa.

Daniel Silva (33.33%): Responsável principal pela implementação dos algoritmos de pesquisa. Co-responsável pelo cálculo empírico da complexidade;

Tomás Oliveira (33.33%): Responsável pela manipulação dos dados de entrada do programa; Co-responsável pelo cálculo empírico da complexidade; Responsável pela implementação dos Menus; Auxiliar na elaboração do relatório.

7. Conclusão

À primeira parte do projeto responsável pela criação de um supermercado ao domicílio foi agora adicionado a capacidade de pesquisa em strings (exata e aproximada).

O programa é agora capaz de identificar supermercados por qualquer uma das vias supracitadas bem como de fornecer sugestões adequadas para as pesquisas falham.

A análise teórica da complexidade dos algoritmos foi comprovada pela análise empírica dos mesmos que surge anexada no apêndice A.

8. Referências bibliográficas

[1] Data Structures and Algorithm Analysis in Java, Second Edition, Mark Allen Weiss, Addison Wesley, 2006

9. Apêndice A