

Faculdade de Engenharia da Universidade do Porto



Bike Sharing – Parte 2

Francisco Tuna Andrade - up201503481@fe.up.pt

Nádia de Sousa Varela de Carvalho - ei12047@fe.up.pt

Paulo Renato Almeida Correia - up201406006@fe.up.pt

Relatório de Projeto realizada(o) no âmbito da disciplina
Conceção e Análise de Algoritmos
Mestrado Integrado em Engenharia Informática e Computação

Regente: Prof. Dr. Rosaldo José Fernandes Rossetti

2017/05/28

Índice

Introdução.....	3
Descrição do Problema	4
Identificação do Problema (Pesquisa Aproximada)	5
Dados de Entrada	5
Restrição	5
Função Objetivo	5
Resultado esperado	5
Formalização do Problema (Pesquisa Aproximada)	6
Dados de Entrada.....	6
Restrição	6
Função Objetivo	6
Resultado Esperado	6
Algoritmos e complexidade	7
Pesquisa exata	7
Pesquisa aproximada	7
Descrição da solução.....	9
Casos de utilização	10
Principais dificuldades.....	11
Esforço dedicado por cada elemento do grupo.....	11
Conclusão.....	12

Introdução

Neste relatório pretendemos abordar os vários tópicos pedidos na descrição do trabalho, nomeadamente: a descrição do tema, a identificação e formalização do problema, a descrição da solução encontrada e a lista de casos de utilização.

Tencionamos também fazer o relato das principais dificuldades encontradas na realização do projeto, a análise do contributo dado por cada um dos elementos do grupo e indicar o nosso nível de satisfação com a solução implementada.

Descrição do Problema

O nosso projeto consiste no desenvolvimento de uma aplicação de partilha de bicicletas para uma cidade. Na primeira parte do projeto construímos um programa responsável por gerir os dados dos utilizadores da aplicação, tais como nome, número de cartão de crédito e saldo corrente e por indicar ao utilizador qual o ponto de partilha de bicicletas mais próximo do local onde ele se encontra.

Na segunda parte, foi-nos incumbida a tarefa de acrescentar, às funcionalidades do programa já existentes, a capacidade de procurar pontos de partilha de bicicletas pelo nome da rua em que se encontram. É necessário disponibilizar duas pesquisas, a pesquisa exata que retorna os pontos que se encontram numa determinada rua escrita pelo utilizador e a pesquisa aproximada que retorna os nomes de ruas mais próximos, ordenados por similaridade, onde existe um ponto de partilha.

Identificação do Problema (Pesquisa Aproximada)

De uma forma geral, pretendemos resolver o problema da pesquisa aproximada através de um programa, que, recebendo uma String, descobre qual os pontos de bicicletas situados em ruas com nomes mais próximos da String fornecida pelo utilizador, retornando-os por ordem crescente de proximidade.

Dados de Entrada

- *Vector < Road >*
- *String Inicial*

Road – classe que contem o nome de uma rua
e os vetor de pontos de partilha nela contidos

Restrição

A distância entre os nomes das ruas pontos de partilha retornados ao utilizador e a String input deve ser menor ou igual a 20.

Função Objetivo

Facilitar a pesquisa de locais onde podem ser deixadas as bicicletas.

Resultado esperado

Vetor com os pontos situados em ruas com nome semelhante àquela que digitada pelo utilizador, ordenadas por ordem crescente de proximidade.

Formalização do Problema (Pesquisa Aproximada)

De seguida mostra-se a formulação matemática do problema:

Dados de Entrada

- $Vector < Road >: R$
- $String Inicial: I$

Restrição

$$\forall p \in Approx_{Points} \text{ dist}(p.name, I) \leq 20$$

Função Objetivo

$$Approx_{Points} = \{p \mid \text{dist}(p.name, I) \leq 20\} \text{ e}$$

$$p_i, p_j \in Approx_{Points} \wedge i < j \Rightarrow \text{dist}(p_i.name, I) \leq \text{dist}(p_j.name, I)$$

Resultado Esperado

$$Approx_{Points} = \{p\}, \text{ onde } p \text{ é ponto aproximado}$$

Algoritmos e complexidade

Pesquisa exata

Para a pesquisa exata foi usado um algoritmo que compara a String fornecida pelo utilizador com os nomes das ruas caractere a caractere e retorna os pontos de partilha que pertencem a uma rua com o mesmo nome da String fornecida pelo utilizador ou retorna um vetor vazio caso não exista nenhuma rua com o mesmo nome que aquele que foi fornecido pelo utilizador. Sendo assim, a complexidade deste algoritmo é, em média, $O(|I| * \text{numero_ruas} * \text{numero_medio_pontos_por_rua})$, onde I é a String dada como input pelo utilizador.

Empiricamente podemos confirmar que este é o valor da complexidade do algoritmo, utilizando o mapa da área FEUP com um total de 854 ruas e um número médio de 5 pontos por rua e o mapa da região metropolitana de Lisboa com 12886 ruas e um número médio de 5 pontos por rua. Em ambos casos, o tempo que este algoritmo demorou a ser aplicado foi desprezável, menos de meio segundo.

Pesquisa aproximada

Para a pesquisa aproximada foi utilizado um algoritmo que calcula a distância entre cada rua e a String fornecida pelo utilizador, retornando os pontos de partilha situadas nas ruas com uma distância à String fornecida pelo utilizador menor ou igual a 20. Ou seja, encontrando as ruas com uma distância menor ou igual a 20, o algoritmo precisa ainda de procurar os pontos nessa rua que são pontos de partilha e de seguida ordenar esses pontos por ordem crescente de distância à String do utilizador. Considerando que o algoritmo que calcula a distância entre o nome de cada rua e a String fornecida I tem complexidade $O(|I| * |\text{rua fornecida}|)$, a pesquisa aproximada tem uma complexidade de $O(|I| * \text{numero_ruas} * \text{numero_medio_pontos_}$

por_rua*tamanho_medio_nome_de_rua).

Empiricamente podemos confirmar que este é o valor da complexidade do algoritmo, utilizando o mapa da área FEUP, com um total de 854 ruas e um numero médio de 5 pontos por rua e o mapa da região metropolitana de Lisboa com 12886 ruas e um número médio de 5 pontos por rua. O tamanho médio do nome de uma rua em cada caso é 30. No mapa da Área da FEUP, este algoritmo executa sempre em menos de um segundo, enquanto que no mapa da região metropolitana de Lisboa com Strings mais pequenas o tempo que o algoritmo demora a executar é desprezável, mas com Strings maiores este algoritmo chega a demorar até 2 segundos.

Descrição da solução

O nosso projeto permite obter informações acerca de duas zonas distintas: a área ao redor da FEUP e a região metropolitana de Lisboa. Alguns funcionalidades já utilizadas na parte 1 deste trabalho foram melhoradas, enquanto que outras foram adicionadas, como é o caso da pesquisa de ruas.

Para obter informações acerca das cidades pretendidas decidimos recorrer aos mapas disponibilizados no site Open Street Maps, que contém informação acerca dos pontos de uma cidade, das suas ligações rodoviárias e duas ruas em que estes se encontram. Depois de termos colocado estes dados num grafo, extraímos a sua maior componente conexa para que não existissem locais da cidade isolados e construímos assim o grafo usado na resolução do problema. A informação acerca das ruas foi colocada num vetor em que cada elemento contém o nome de uma rua e os ID's dos pontos que fazem parte dessa rua.

A leitura do grafo de cada vez que o programa é iniciado é feita utilizando um ficheiro com o seu id de todos os vértices e as suas coordenadas geográficas e outro ficheiro com o id de origem de cada uma das arestas, assim como o id de destino. Para fazer a leitura do grafo é necessária uma complexidade temporal de $O(\text{arestas} \cdot \text{vértices})$.

Para fazer a pesquisa exata de ruas foi utilizado um simples algoritmo de comparação de Strings e para a pesquisa aproximada de ruas foi utilizado o algoritmo vulgarmente conhecido como “approximate string matching”. Os pormenores relacionados com a implementação destes algoritmos já foram discutidos acima.

Casos de utilização

- Construção de um grafo através do Open Steet Maps
- Leitura de ficheiros com informações acerca do grafo construído
- Programa que faz o registo e login de utilizadores
- Programa que lida com o saldo disponível dos utilizadores, não permitindo levantar bicicletas quando estes não tiveram saldo
- Escolha do ponto de partilha mais próximo da localização atual do utilizador
- Visualização, através do GraphViewer, de um mapa área em redor da FEUP, onde se mostram os pontos de partilha de bicicletas
- Visualização, através do GraphViewer, de um mapa em torno da região metropolitana de Lisboa, onde se mostram os pontos de partilha de bicicletas
- Visualização, através do GraphViewer, do percurso real entre o utilizador e o ponto de partilha mais próximo
- Pesquisa exata de nomes de ruas onde existem pontos de partilha de bicicletas
- Pesquisa aproximada de nomes de ruas onde existem pontos de partilha de bicicletas.

Principais dificuldades

Neste trabalho, a nossa maior dificuldade não se relacionou com a implementação das novas funcionalidades de pesquisa de ruas, mas sim com a adaptação das funcionalidades implementadas na parte 1 a um mapa maior, que permitisse testar a pesquisa de ruas aproximadamente. Foi-nos particularmente difícil a construção de um grafo correspondente à zona metropolitana de Lisboa, usando os dados fornecidos pelo Open Street Maps, uma vez que estes eram muito grandes.

Esforço dedicado por cada elemento do grupo

Todos os elementos do grupo discutiram a solução a implementar e colaboraram uns com os outros na realização do trabalho, sendo que nesta parte 2 do trabalho, coube um papel maior ao estudante Francisco Andrade, responsável pela implementação das novas funcionalidades e da adaptação dos algoritmos usados na parte 1 para mapas maiores.

Conclusão

Pensamos ter sido bem-sucedidos na realização deste trabalho, tendo cumprido todos os objetivos pedidos e acabando o projeto no prazo indicado. cremos que houve colaboração entre os elementos do grupo, o que contribuiu para que as dificuldades que apareceram fossem ultrapassadas.

Apenas ficamos desiludidos com o facto de não termos conseguido encontrar uma solução mais rápida para a pesquisa aproximada.