07-04-2017



Beatriz Souto de Sá e Baldaia up201505633@fe.up.pt Daniel Pereira Machado up201506365@fe.up.pt Nelson André Garrido da Costa up201403128@fe.up.pt Sofia Catarina Bahamonde Alves up201504570@fe.up.pt

FEUP

RELATÓRIO DE CAL

À Procura de Estacionamento

ÍNDICE

INTRODUÇÃO

O objetivo deste trabalho é dar resposta ao problema de procurar estacionamento numa determinada cidade, ou seja, conseguir auxiliar o utilizador na procura de um lugar de estacionamento. Por exemplo, o utilizador pretende deslocar-se até ao supermercado, o sistema deve ser capaz de lhe calcular o itinerário para o estacionamento (de automóvel) e do estacionamento até ao supermercado (a pé). Se a rede de estradas for representada por um grafo, uma possível abordagem ao problema será encontrar o lugar mais próximo do destino, ou também poderá passar por encontrar o lugar mais barato dentro de uma distância máxima.

Descrição do Problema

Nos dias de hoje, qualquer cidadão lida com problemas relativos ao local onde poderá estacionar o seu automóvel, numa viagem pela sua cidade.

Neste trabalho, pretende-se através de um sistema auxiliar o utilizador na procura de um local para estacionar mais oportuno de acordo com o seu trajeto. Com oportuno, referimo-nos a calcular o itinerário mais eficaz (mais curto e rápido) desde um certo local de origem até a um de destino (cinema, escola, shopping, etc.), tendo em conta o caminho até ao estacionamento, que pode ser em estrada (parquímetro) ou de garagem.

É inquirido ao utilizador que especifique determinados detalhes:

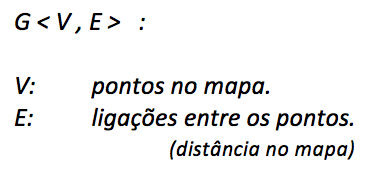
* Local de origem
* Local de destino
* Distância máxima a pé desde o local de estacionamento ao destino
* Se pretende abastecer o seu automóvel durante a viagem

Solução Apresentada

De uma forma geral, pretendemos resolver o problema apresentado, recebendo o local de origem e o local de destino, de modo a devolver o caminho mais eficaz, de acordo com as preferências do utilizador.

# Input

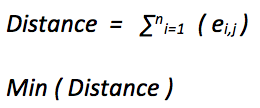
Construção de um grafo, G = (V, E), de pontos e ligações entre os pontos no mapa da cidade, no qual:

* G – Representa o grafo de ligações entre os diversos pontos
* V – Representa o conjunto de nós (ou vértices), ou seja, pontos (de interesse ou não) no mapa
* E – Representa o conjunto de arcos (ou arestas), ou seja, as ruas do mapa em questão
* vi – Representa um nó (ou vértice)
* (xi, yi) – Representa as coordenadas x,y do nó vi
* ei,j – representa o arco entre o nó vi  e o nó vj
* di,j - Representa a distancia do arco ei,j

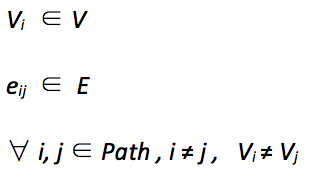
# Output

# Estrutura de dados contendo os pontos (vértices) do grafo (mapa) pelos quais o automobilista deverá passar, de modo a percorrer o caminho mais curto.

# Objetivo

Apresentar ao utilizador o caminho mais favorável, consoante as preferências apresentadas por este, e de acordo com o calculado.

# Restrições



Os vértices do caminho pertencem ao conjunto V.  
As arestas que representam o caminho pertencem ao conjunto E.  
Não pode passar no mesmo vértice mais que uma vez.

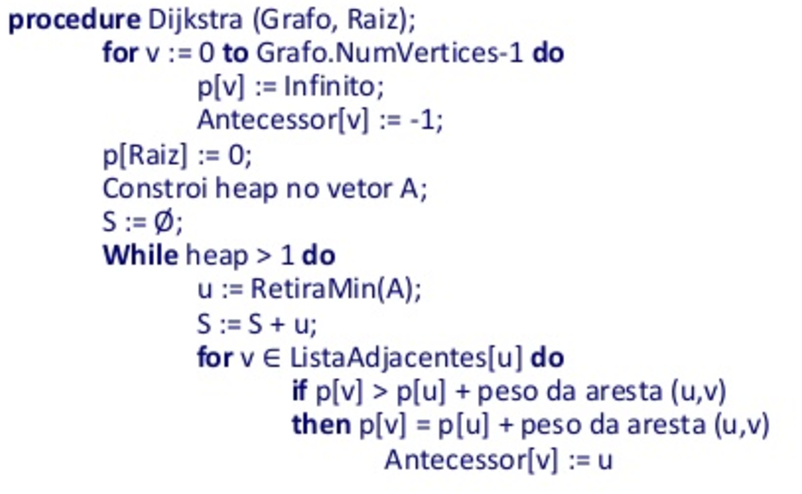
# Algoritmo

Consideramos o ***algoritmo de Dijkstra***o mais conveniente nesta solução.

Este encontra o caminho mais curto num grafo (dirigido ou não dirigido) em tempo computacional **O([arestas + vértices]log (vértices))**.

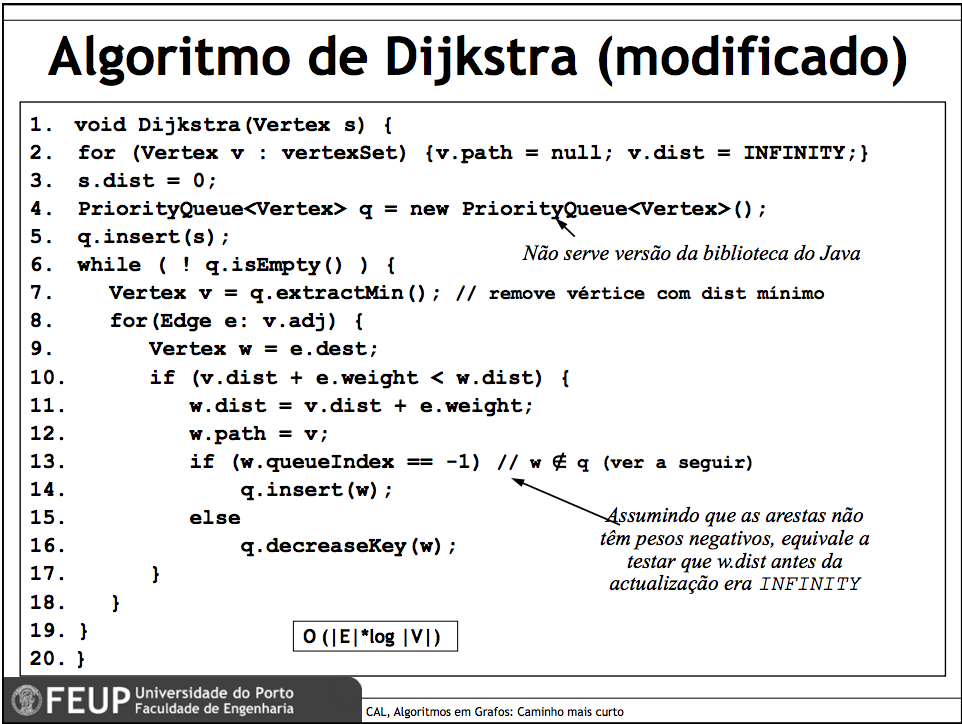
Trata-se de um algoritmo ganancioso (*greedy*) - *um problema tem subestrutura ótima se uma solução ótima para um problema contém soluções ótimas para os seus subproblemas.*

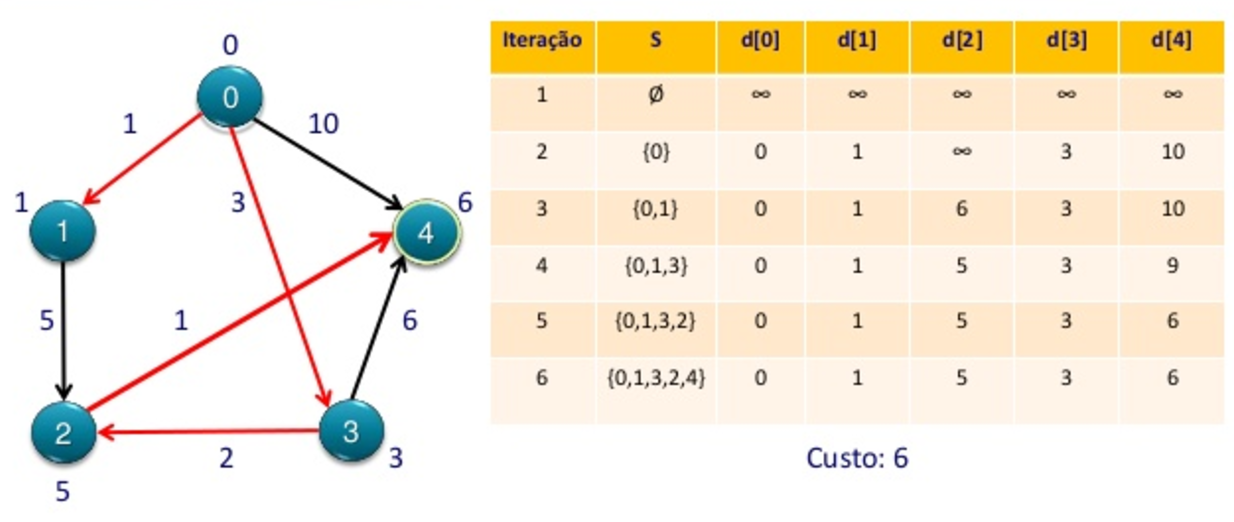
O valor (peso) de cada arco está associado à distância, que nunca será negativo, pelo que este algoritmo é aplicável neste problema.



Algoritmo de Dijkstra (modificado) - pseudocódigo

Algoritmo de Dijkstra (modificado) – slides CAL 2017





Cálculo do caminho mais curto do vértice 0 ao vértice 4, pelo algoritmo de Dijkstra

# Esquema da Solução

1. Introdução dos dados do grafo (vértices – pontos no mapa, e arestas – arcos entre os pontos)
2. Diferenciação do grafo:
   1. Representação gráfica (*GraphViewer*).
   2. Estrutura de auxílio na implementação.
3. Pedido de informação ao utilizador:
   1. Rua de origem (consola).
   2. Apresentação da rua com todos os seus vértices do grafo, de forma a que o utilizador escolha qual o vértice mais perto do seu local de origem (mapa).
   3. Apresentação dos locais de interesse, para que o utilizador escolha para onde quer ir – definição do vértice de destino (mapa).
   4. Saber que tipo de caminho pretende: o mais rápido ou o mais barato (consola).
   5. Saber se pretende abastecer o automóvel durante o percurso.
   6. Apresentação no mapa do caminho selecionado pelo algoritmo, assim como o lugar de estacionamento selecionado (mapa).

Falta: imagens

# Complexidade

**O([arestas + vértices]log (vértices))**

Falta: mostrar analiticamente este valor.