

Bárbara Silva  **up201505628**@fe.up.pt

João Azevedo **up201503256**@fe.up.pt

Julieta Frade **up201506530**@fe.up.pt

Conceção e Análise de Algoritmos

*À Procura de Estacionamento*

*2MIEIC02 – Grupo D*

*(9 de abril de 2017)*

Índice

|  |  |
| --- | --- |
| **Introdução** | 2 |
| Descrição do Tema | 2 |
| Identificação e Formalização do Problema | 3 |
| **Solução Implementada** | 4 |
| **Diagrama de Classes** | 6 |
| **Casos de Utilização** | 7 |
| Classes | 7 |
| Ficheiros | 7 |
| Algoritmos | 8 |
| Programa | 9 |
| **Dificuldades** | 10 |
| **Distribuição do Trabalho pelos Elementos do Grupo** | 11 |
| **Conclusão** | 12 |
|  |  |

***Introdução***

*Descrição do Tema –* “À procura de estacionamento”

Este projeto realizado no âmbito da unidade curricular de CAL destina-se ao processamento de um mapa real, neste caso uma área por nós escolhida *à priori,* e respetivo cálculo do caminho ideal de um local origem para um destino, ambos indicados pelo automobilista.

No contexto do problema, o “caminho ideal” assume diferentes semânticas, uma vez que, por exemplo, o caminho pretendido pode ser o que nos leva ao destino pelo caminho mais curto ou que nos leva ao parque mais barato.

Assim, identificamos como objetivos principais do trabalho os seguintes:

* Possibilidade do utilizador definir o local de origem e destino
* Possibilidade de escolher o caminho:
  + Mais curto até ao parque associado ao destino
  + Que leva ao parque mais barato associado ao destino
* Possibilidado de escolher qualquer uma das opções acima desejando abastecer o veículo ou não

Todos os parques por nós indicados ao utilizador estarão a uma distância máxima do local de destino, distância essa definida pelo utilizador.

*Identificação e Formalização do Problema*

Enquanto grupo de trabalho procurámos desde cedo ler e reler o enunciado proposto e formalizar o problema, subdividindo-o em subproblemas até que se tornassem elementares e de simples resolução.

Assim, identificamos e modulámos o nosso projeto em alguns tópicos:

1. A escolha de uma área real de teste
2. O uso do parser providenciado
3. A extração de informação dos ficheiros de texto
4. A criação da estrutura de dados a partir dos mesmos ficheiros:
   * Criação de nós que contivessem a informação pretendida
   * Criação de arestas com os valores essenciais aos cálculos futuros
   * Criação conjunta do grafo, englobando nós e arestas
5. A determinação do caminho ideal no grafo usando o algoritmo adequado
6. A criação de um menu que serviria de interface com o automobilista

***Solução Implementada***

O primeiro passo foi então definir a área a testar e se para tal usaríamos o parser providenciado ou nós próprios criarimos os ficheiros contendo a informação necessária. Rapidamente nos apercebemos que seria mais prático e elucidativo criarmos os nossos próprios ficheiros de teste.

Efetivamente, tendo os ficheiros prontos e uma ideia bem definida da área a utilizar, facilmente criámos as funções necessárias à leitura de ficheiros de texto, criando também os nós e arestas do nosso grafo.

Cada nó possui atributos necessários ao funcionamento dos algoritmos à frente explicados, como se já foi visitado pelo mesmo. Em cada nó está também armazenada a lista de arestas a que o vértice(nó) está ligado. Por último, a informação útil em si, contida num objeto template por nós definido contendo o seu ID, o seu nome, indicativo do local (cinema,hospital,...), e um par de coordenadas no mapa (x,y).

Os parques são objetos que contêm os nós a eles associaods, isto é, um parque está associado a diferentes destinos (próximos do mesmo). Para além disso, guardam o seu nome e preço e ainda a informação de se é um parquímetro ou garagem.

As arestas são as ruas em si e contêm não só um ID e o nome da mesma mas também a informação sobre a existência de dois sentidos ou apenas um e uma lista de nós que liga.

Para o cálculo do caminho mais curto, uma vez tratar-se de um grafo pesado com apenas números inteiros positivos, foi usado o algoritmos de Dijkstra.

***Diagrama de Classes***

No fim do trabalho fazer UML disto em 5 mins e colar aqui

***Casos de Utilização***

*Classes*

Manager 🡪 Singleton class (instância da mesma limitada a um objeto) que contem a informação do programa em execução assim como os metodos para o iniciar

Node 🡪 classe representativa de um local(nó), contendo um ID, o nome desse local e um par de coordenadas no mapa

Street 🡪 classe representativa de uma estrada(aresta), contendo um ID, o nome da rua, um boleano indicando se é ou não de dois sentidos e uma lista de nós que a estrada liga

ParkingLot 🡪 classe representativa do parque de estacionamento, contendo um ID, o nome do mesmo, o preço a pagar, um boleano indicando se é ou não garagem (caso não o seja é um parquímetro) e uma lista de nós ao qual o parque se associa, ou seja, nós do qual o parque é considerado perto o suficiente para ser uma boa opção

*Ficheiros*

Edges.txt – ficheiro contendo a informação relativa a arestas como o seu ID e os nós que liga

Nodes.txt – ficheiro contendo o ID do nó, as coordenadas x e y no mapa e o nome identificativo do local

Parking.txt – ficheiro contendo

Streets.txt – ficheiro contendo a informação das ruas(arestas), como o seu ID, o nome da mesma

*Algoritmos*

*Algoritmo de Djikstra*

Para a realização do projeto utilizámos apenas um algoritmo, sendo um tema leccionado na cadeira, para a determinação do caminho mais curto num grafo pesado com valores apenas positivos. Falamos pois do algoritmo de D jikstra.

Em termos de analise de complexidade espacial e temporal este algoritmo apresenta:

Espacialmente: O( N^2 ) onde N representa o número de nós

Temporalmente: O( A + N log(N) ) onde N representa o número de nós e A o de arestas

Este algoritmo é responsável pelo cálculo do caminho mais curto do local de partida ao destino. É usado tanto para a determinação do caminho mais curto como do caminho que leva ao parque menos custoso em redor do destino. Faz parte do cálculo ainda dos trajetos:

1. Partida 🡪 Destino
2. Partida 🡪 Abastecimento 🡪 Destino

|  |  |  |
| --- | --- | --- |
| V | Arestas | T(ms) |
| 26 | 43 | < 1 |
| 200 | 100 | 7 |
| 600 | 300 | 18 |
| 1000 | 500 | 28 |

Inserir pesudocodigo.

Inserir grafico e conclusao sobre dados acima

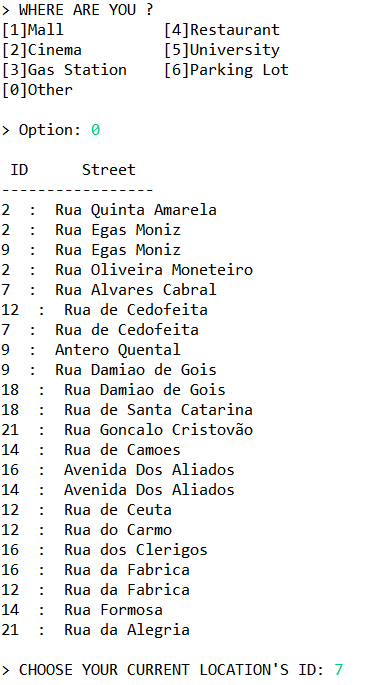
Algoritmo da conetividade

*Programa*

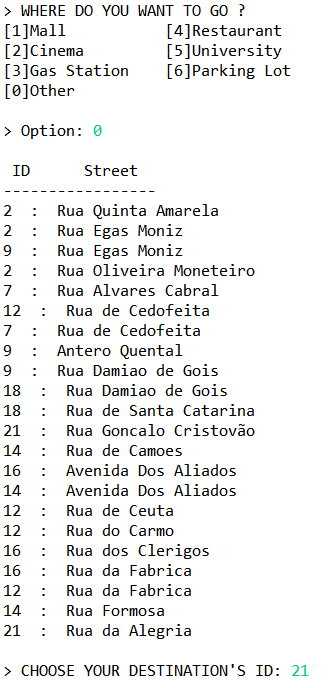
Primeiramente, o programa executa o processamento da informação dos ficheiros de texto, respetiva criação do grafo e visualização do mesmo com toda a informação identificadora de cada aresta e nó.

Numa segunda fase, é requerida informação ao automobilista através da consola.

Para começar, é requerida a posição atual do utilizador, como abaixo indicado:



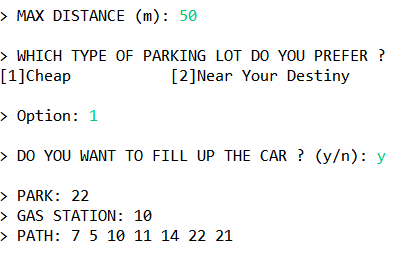
De seguida é necessário, tal como abaixo demonstrado, indicar o destino pretendido:



Por fim, a distância máxima do local destino ao parque de estacionamento deve ser indicada, assim como a especificação do caminho ideal, podendo este ser o mais curto ou o que conduz ao parque mais barato dentro da distância máxima especificada ao destino.

Como parâmetro adicional, o automobilista pode, para ambas as opções anteriormente especificadas, optar por um trajeto que inclua um posto de abastecimento ou não.

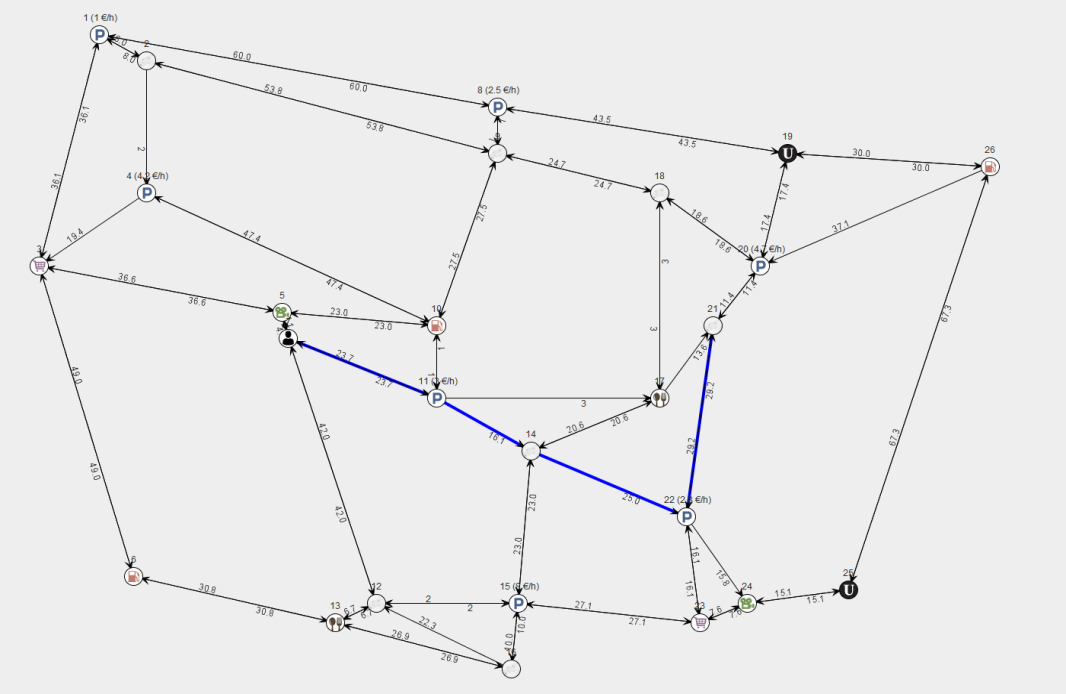
A imagem seguinte ilustra então o resultado final, indicando-se o ID do parque ideal e o caminho a percorrer, nó a nó, e, se pretendido, o posto de abastecimento automóvel onde parar.



De seguida, apresentámos alguns dos resultados visualizados mediante cada um dos quatro casos possíveis.

Partida 🡪 Destino mais barato

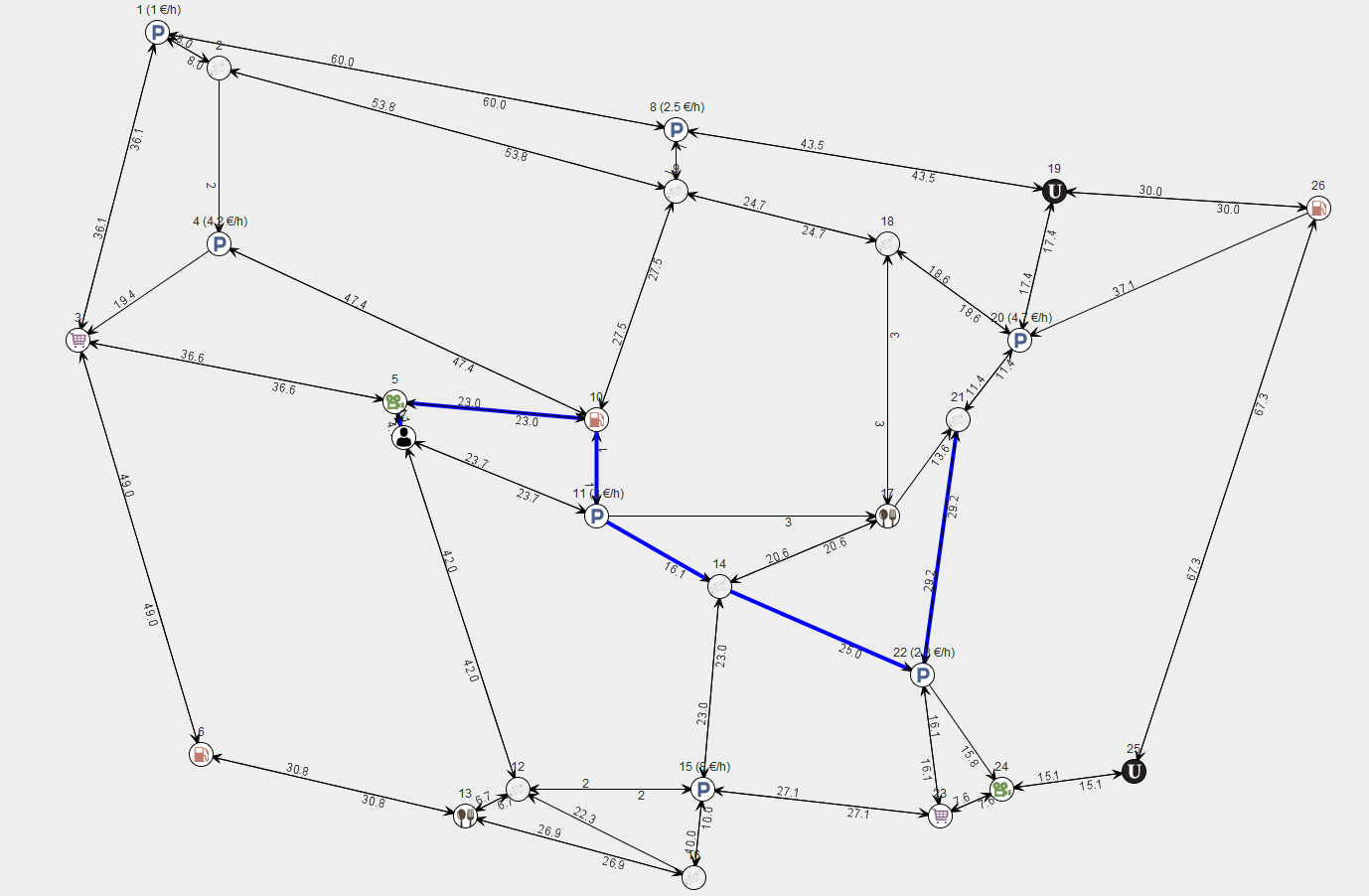
Caso se pretenda estacionar no parque mais barato perto do destino e não se faça questão de abastecer, como por exemplo, fazendo este mesmo percurso entre os nós 7 e 21, o resultado visualizado será o seguinte:



É de notar que o caminho ideal calculado se encontra delineado a azul.

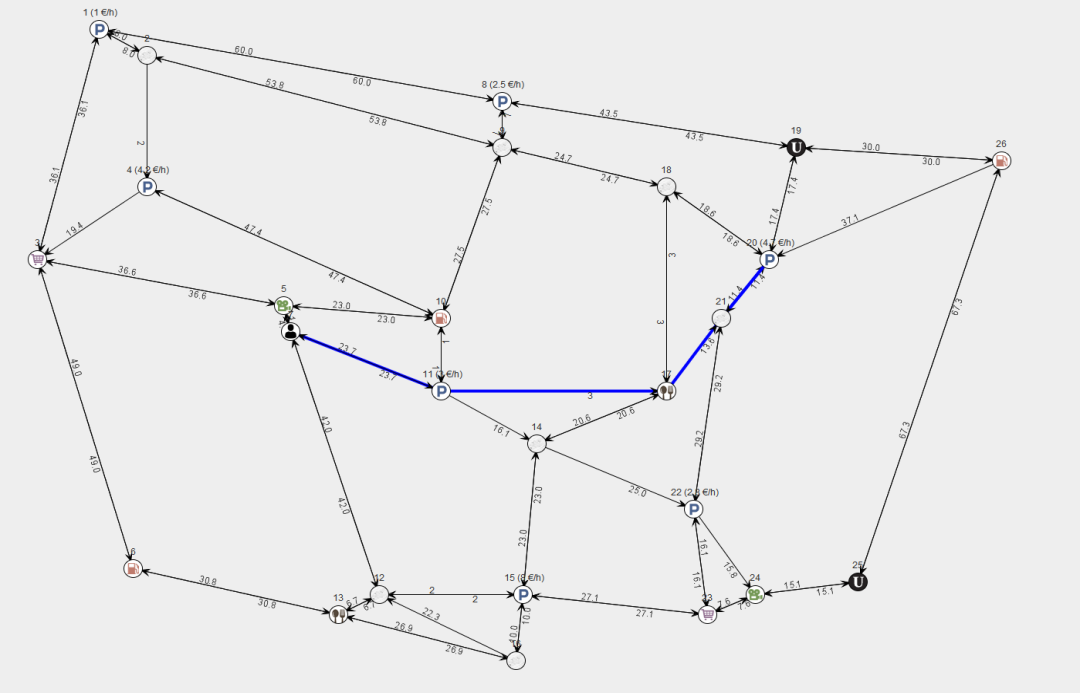
Partida 🡪 Posto Abast. 🡪 Destino mais barato

Ainda trabalhando sobre os nós 7 e 21, podemos visualizar o seguinte resultado quando pretendido o caminho que leve ao parque mais barato perto do destino, passando pelo posto de abastecimento.



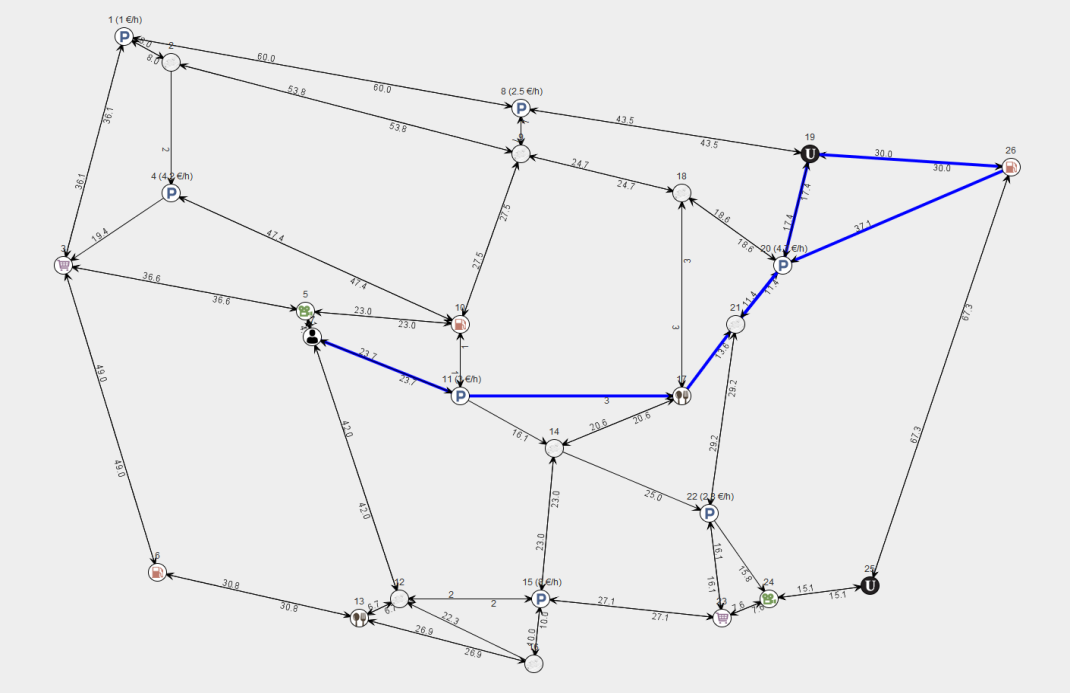
Partida 🡪 Destino pelo caminho mais curto

Abaixo é apresentado o resultado de optar por não querer abastecer o veículo e pretender colocar o parque o mais perto possível do local destino.



Partida 🡪 Posto Abast. 🡪 Destino pelo caminho mais curto

Por fim, caso o automobilista pretenda o caminho mais curto do local em que se encontra ao destino, passando por um posto automóvel, referindo-nos ainda como exemplo aos nós 7 e 21, obtemos o seguinte grafo:



***Dificuldades***

Durante a realização deste projeto não se apresentaram grandes dificuldades a nível de programação em si mas sim de edificação e planeamento do mesmo.

Efetivamente, provavelmente o maior obstáculo terá sido a compreensão da utilidade do software apresentado (GraphViewer e Parser) e de como utilizá-lo.

Tal como referido, optámos por descartar o parser por considerarmos desnecessário e não benéfico mas usámos o GraphViewer para visualização do nosso grafo.

Ainda relativamente a este ponto, o setup do projeto em diferentes IDE’s (Visual Studio e Eclipse) demorou talvez um pouco mais do que seria esperado.

***Distribuição do Trabalho pelos Elementos do Grupo***

Apesar do esforço de todos os envolvidos na compreensão do enunciado assim como na modulação do problema e tomada de decisão a nível de funcionamento do programa, a distribuição do trabalho nao foi equitativa, podendo-se estimar as seguintes percentagens e distribuição de tarefas:

- Julieta Frade (40%): ESCREVE O QUE FIZESTE

- Sofia Silva (30%): ESCREVE O QUE FIZESTE

- João Conde (30%): Redação do relatório

***Conclusão***

A proposta de trabalho possuía um intuito educativo, sendo requerido da nossa parte que comprendessemos e usassemos não só novas estruturas como grafos mas também algoritmos de pesquisa nos mesmos. Não só, mas também o desenvolvimento da componente de trabalho em grupo.

Concluimos portanto que os objetivos pretendidos com este projeto de grupo foram atingidos, quer a nível individual quer a nível coletivo, uma vez que cada elemento domina agora os temas leccionados na unidade curricular e é capaz de os aplicar numa componente prática.