WasteApp: recolha seletiva de lixo (Parte 1)

Turma 1 - Grupo 1

up201806551@fe.up.pt	Beatriz Costa Silva Mendes
up201806524@fe.up.pt	Daniel Garcia Lima Sarmento da Silva
up201806538@fe.up.pt	Henrique Manuel Ruivo Pereira

24 de abril de 2020

Projeto CAL - 2019/20 - MIEIC

Professora das Aulas Laboratoriais: Liliana da Silva Ferreira

Índice

1 Descrição

Neste trabalho pretende-se criar uma aplicação para gestão e localização dos pontos de recolha seletiva de resíduos, denominada *WasteApp*. Esta *app* deverá permitir aos seus utilizadores localizar os pontos de recolha seletiva mais próximos e os tipos de resíduos que lá se podem depositar, bem como realizar a gestão da sua capacidade.

Para além disso, a *app* tem ainda por objetivo dar origem a um modelo de negócio que tem por base a recolha ao domicílio de determinados tipos de resíduos para exploração financeira (otimizando o itinerário percorrido).

A aplicação deve ser capaz de determinar a acessibilidade aos pontos de depósito/recolha.[1]

1.1 1ª Abordagem: Ponto de Recolha Mais Próximo

Numa fase inicial, despreza-se a capacidade do ponto de recolha e os tipos de resíduos que poderão ser depositados neste. Deste modo, o único objetivo acaba por ser apenas determinar qual o ponto de recolha mais próximo do utilizador, calculando a rota mais curta até um qualquer ponto de recolha (partindo-se do princípio de que este se encontra acessível).

1.2 2ª Abordagem: Ponto de Recolha de um Determinado Resíduo Mais Próximo

Neste segundo passo, acrescenta-se ao problema o facto de que determinados pontos de recolha se limitam a aceitar certos tipos de resíduos. Assim, cada utilizador terá de ter pelo menos 5 pontos de recolha mais próximos, um para cada tipo de resíduo (papel, vidro, plástico, pilhas e lixo indiferenciado).

1.3 3ª Abordagem: Ponto de Recolha de um Determinado Resíduo Mais Próximo com Capacidade Suficiente

Posteriormente, é necessário considerar que os pontos de recolha têm uma capacidade máxima, ou seja, depois de atingir essa capacidade, não poderá ser depositado nele mais nenhum resíduo. Assim sendo, se o ponto de recolha mais próximo do utilizador de um determinado resíduo ultrapassar a sua capacidade máxima com o depósito do utilizador, terá de ser atribuído a este um novo ponto de recolha desse mesmo resíduo, que será o mais próximo ainda com capacidade.

1.4 4ª Abordagem: Implementação Paralela do Modelo de Negócio

Por fim, neste última abordagem, terá de ser implementado um serviço de recolha de resíduos ao domicílio para exploração financeira, que depois serão levados para uma central de reciclagem. Assim sendo, terá de ser determinado o menor itinerário possível que passe pelas casas dos utilizadores que fornecem os resíduos a ser recolhidos.

2 Identificação e formalização do problema

3 Perspetiva de Solução

A perspetiva de solução adotada tem por base a aplicação de várias fases prévias ao processamento do problema.

Inicialmente, e para reduzir a complexidade temporal do processamento, os passos iniciais baseiam-se na eliminação de arestas que não proporcionem nenhum tipo de vantagem (tais como as correspondentes a vias inutilizáveis - por obras, por exemplo - e as que tenham arestas/conjuntos de arestas com origem e destino comuns de peso total menor).

O passo seguinte corresponde à ordenação dos pontos de recolha por proximidade a cada utilizador. Tendo em conta que a morada de um utilizador e os pontos de recolha podem corresponder a um nó ou a uma parte de uma aresta, se se tratar do primeiro caso apenas se tem de ter em conta as arestas adjacentes; quanto ao segundo caso, terá de se localizar o ponto de recolha ou a morada dentro da aresta, isto é, saber a distância deste(a) a cada uma das extremidades da aresta.

Para o primeiro caso, usa-se o algoritmo de *Dijsktra* abordado nas aulas, cujo pseudocódigo é apresentado na figura seguinte[2], uma vez que o grafo não contem arestas de peso negativo.

```
DIJKSTRA(G, s):  // G=(V,E), s ∈ V
1.    for each v ∈ V do
2.       dist(v) ← ∞
3.       path(v) ← nil
4.    dist(s) ← 0
5.    Q ← ∅   // min-priority queue
6.    INSERT(Q, (s, 0)) // inserts s with key 0
7.    while Q ≠ ∅ do
8.       v ← EXTRACT-MIN(Q) // greedy
9.       for each w ∈ Adj(v) do
10.         if dist(w) > dist(v) + weight(v,w) then
11.            dist(w) ← dist(v)+ weight(v,w)
12.            path(w) ← v
13.            if w ∉ Q then  // old dist(w) was ∞
14.               INSERT(Q, (w, dist(w)))
15.            else
16.               DECREASE-KEY(Q, (w, dist(w)))
```

Tempo de execução:
O((|V|+|E|) * log |V|)

A utilização deste algoritmo de cálculo do caminho mais curto desde o vértice inicial (morada do utilizador) até aos outros nós (pontos de recolha) resulta numa árvore de caminhos ordnad.

A preparação do algoritmo tem complexidade temporal O($|V|$) e consiste na inicialização do campo **dist** a ∞ e **path** a "nil"(correspondente a *nullptr*) em todos os vértices (à exceção da origem). Posteriormente, a partir do vértice inicial, percorre-se todas as arestas adjacentes e adiciona-se os vértices de destino a uma fila de prioridade, atualizando a dist (distância do vértice anterior somada com o peso da aresta) e o path (vértice anterior), se este vértice ainda não tiver sido visitado, ou se a distância obtida for menor do que o valor prévio de dist.

Este processo tem complexidade temporal O($(|V| + |E|) * log(|V|)$), uma vez que a cada passo - O($|V| + |E|$) - pode ser necessário adicionar, remover ou mover elementos na fila de prioridade - O($log(|V|)$).

Para o segundo caso, ter-se-ia de aplicar uma "translação"ao mesmo algoritmo (uma vez que não se teria de calcular a distância de um nó a todos os outros, mas de um potencial ponto interno da aresta até um potencial ponto interno de outra aresta), tendo que, para todas as distâncias calculadas a partir de cada uma das extremidades dessa aresta (pelo algoritmo referido), somar-se a distância do ponto interno a essa mesma extremidade.

4 Casos de Utilização

A aplicação *Wasteapp* que vamos implementar terá como base uma interface simples de texto com a qual o utilizador poderá interagir. Deste modo, para facilitar a interação, esta terá um conjunto de menus com várias opções a serem disponibilizadas, dividindo-se em 2 menus gerais: 1 menu para o utilizador comum que procura os pontos de recolha mais próximos e 1 menu para o utilizador que irá explorar financeiramente o novo modelo de negócio.

5 Conclusão

6 Referências Bibliográficas

Grupo B, Turma 6, 2016. *Ecoponto: Recolha De Lixo Seletiva (Tema 4) - Parte 1.* Projeto de Concepção e Análise de Algoritmos.

[1] Docs.google.com. 2020. 2019-20 2S CAL Trabalhos Publicados. Disponível no Moodle.

[2] Rossetti, R., Ferreira, L., Cardoso, H. L. e Andrade, F., 2020. *Algoritmos Em Grafos: Caminho Mais Curto (Parte I).*