Distribuição de telecomunicações

Conceção e Análise de Algoritmos

27 de abril de 2015

Turma 2/Grupo A

Francisco Rodrigues - up201305627
Luís Oliveira - up201304515
Miguel Pereira - up201305998

Índice

1. Introdução

No âmbito da unidade curricular de Conceção e Análise de Algoritmos foi-nos proposto um problema onde seria necessário analisar o mapa de uma determinada localidade – representado por um grafo – e distribuir, de forma ideal, a rede de fibra ótica, garantindo cobertura em todas as casas existentes.

Na primeira fase do problema, a fibra deve partir da central e ser instalada ao longo das ruas, de forma a chegar a todas casas; na segunda fase, tendo em conta um número máximo de centrais, deve ser encontrada a melhor forma de as distribuir no terreno.

Neste relatório, será explicado o contexto do problema, bem como a sua formalização e será feita uma descrição da solução, indicando os principais algoritmos implementados.

2. Definição do problema

2.1. Descrição do problema

É pretendido disponibilizar fibra ótica a um conjunto de casas numa determinada localidade. A fibra deve partir de uma central e ser instalada ao longo das ruas da localidade. O objetivo é planear a distribuição que minimiza o cabo utilizado, garantindo cobertura em todas as casas.

A primeira tarefa é, tendo a localização de uma central, descobrir a distribuição ideal da fibra. A segunda tarefa é, dado um número máximo de centrais e um custo relativo à construção de uma central, para além de apresentar a distribuição ótima da fibra, descobrir, ainda, a localização ideal para as centrais, ou seja, aquela que minimiza o custo total.

2.2. Formalização do problema

2.2.1. Input

O mapa de uma localidade, representado por um grafo, $G = (V, E)$, onde:

- V – conjunto de cruzamentos e casas;
- $E_{i,j}$ – rua que liga o cruzamento/casa V_i ao cruzamento/casa V_j.

Para a primeira tarefa é dado o índice c do vértice V_c que representa a central.

Para a segunda tarefa é fornecido o número máximo N_c de centrais que podem ser colocadas e o custo de uma central C_c.

2.2.2. Output

Uma lista de ruas, $L = (E)$, que representa a forma ótima de distribuir a fibra ótica, tal que

$$\forall\, x \in L, x \in E$$

A lista não apresenta ruas repetidas,

$$\forall\, x_i, x_j \in L, i \neq j \rightarrow x_i \neq x_j$$

Sendo C o subconjunto de vértices que representam casas,

$$C \subseteq V$$

tem de existir pelo menos uma rua pertencente a L que chegue a cada um dos elementos de C. Portanto, sendo $x_{i,j}$ a rua que liga V_i a V_j,

$$\forall\, y \in C, \exists\, x_{i,j} \in L : \; y = V_i \; \vee y = V_j$$

2.2.3. Função objetivo

Minimizar o custo total C_t que é dado pela soma do comprimento de cada rua pertencente a L e o número de centrais n utilizadas a multiplicar pelo custo de uma central.

$$\min(C_t) = \min(\sum_{i=1}^{n} comprimento(x_i)\ , x_i \in L + \ n \times C_c, n \ \leq \ N_c)$$

2.2.4. Restrições

Caso existam regiões onde não haja acesso por ruas, então tem de existir pelo menos uma central por cada uma das regiões “isoladas”.

3. Solução implementada

De maneira a resolver o problema proposto, foi necessário combinar dois algoritmos diferentes: o algoritmo de Floyd-Warshall, baseado em programação dinâmica, que descobre o caminho mais curto entre todos os pares de vértices; e o algoritmo de Kruskal (com algumas modificações) que descobre a árvore de expansão mínima de um grafo. Recorreu-se a este método uma vez que este era um caso que poucos dos algoritmos já existentes têm em conta: a existência de vértices obrigatórios (casas) e de vértices opcionais (cruzamentos), com a possibilidade, ainda, dos vértices terem um peso.

A solução implementada foi, então, a seguinte:

- Ao carregar o grafo para a aplicação, guardam-se, num vetor, os índices de todos os vértices que são casas;
- Aplica-se o algoritmo de Floyd-Warshall ao grafo, de forma a saber o caminho mais curto entre todos os pares de vértices. Este algoritmo tem uma complexidade temporal de:

$$O(|V|^3)$$

- Cria-se um vetor que contém todas as arestas entre todos os vértices obrigatórios do grafo (se não existir caminho entre dois vértices, o peso será ∞), utilizando o que vem do algoritmo aplicado anteriormente e ordena-se esse vetor ascendentemente pelo peso. Esta operação tem uma complexidade temporal de:

$$O(|C|^2)$$

- Aplica-se o algoritmo de Kruskal tendo em conta o vetor criado anteriormente. Inicialmente, assume-se que cada vértice do grafo é uma central (uma árvore separada) e, portanto, o custo total é dado pela multiplicação do número de vértices pelo custo de uma central. Cada árvore do grafo só pode ter uma central.
  Depois de escolher uma aresta e verificar que esta não forma um ciclo, verifica-se se a remoção de uma central e a adição do peso da aresta reduz o custo total. Se assim for, dá-se o passo e processa-se a próxima aresta. Caso o número de centrais seja maior que o número máximo, o passo é dado mesmo que o custo aumente com a remoção da central. A complexidade temporal é de:

$$O(|E| \log |V|)$$

- No algoritmo anterior tem-se em conta as arestas entre os vértices obrigatórios, no entanto, pode não existir um caminho direto entre esses dois vértices. É necessário, então, para cada passo dado, adicionar ao caminho final o caminho entre os dois vértices da aresta

escolhida. Para cada aresta que se adiciona, é preciso verificar se aresta não está já no caminho, de modo a evitar arestas repetidas. Assim, a complexidade temporal deste passo é:

$$O(|E|^2)$$

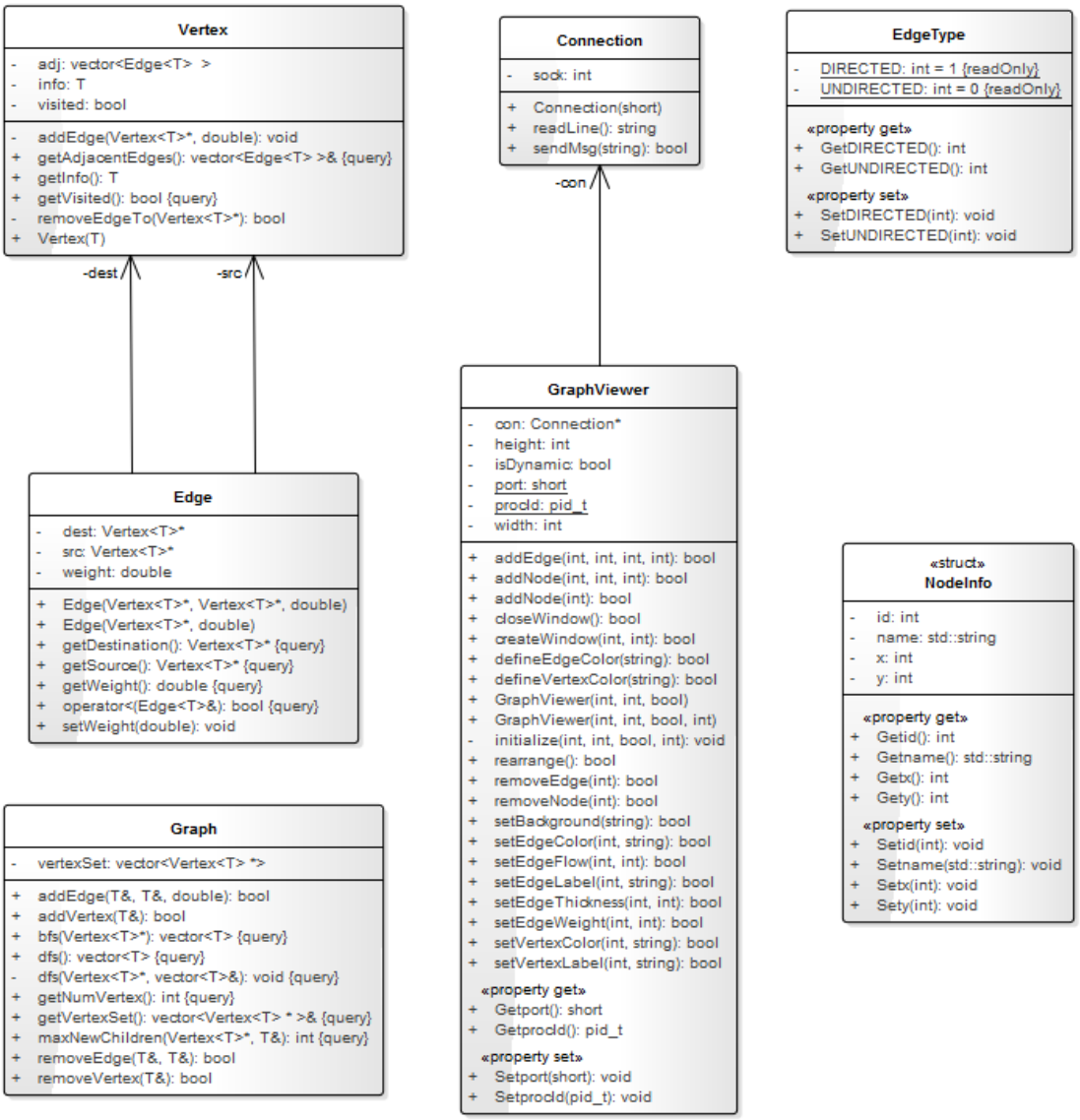
- A cada passo, é preciso, também, atualizar o vetor das arestas. Depois de adicionar as arestas necessárias ao caminho final, o peso de cada uma das arestas que ainda vão ser processadas pode mudar; se o caminho entre dois vértices incluir arestas que já estejam no caminho final, então o peso da aresta entre esses dois vértices tem de diminuir num valor equivalente ao peso dessas arestas. Como para cada aresta se verifica se cada aresta do caminho já existe ou não no caminho final, a complexidade temporal é de:

$$O(|E|^{3)}$$

- Para a tarefa 2 é preciso, ainda, colocar uma central por cada árvore. Percorre-se o vetor do caminho final e, por cada vértice, descobre-se a sua raiz e verifica-se se a árvore com essa raiz já tem uma central. Caso não tenha, coloca-se a central nesse vértice e atualiza-se o vetor das árvores com centrais. No pior caso, A complexidade temporal é de:

$$O(|V|^2)$$

4. Diagrama UML



5. Lista de casos de utilização

- Visualizar um grafo;
- Gerar um grafo aleatório com um dado número de vértices;
- Determinar a distribuição ideal da fibra ótica numa localidade tendo em conta a localização de uma central;
- Determinar a distribuição ideal da fibra ótica e das centrais numa localidade tendo em conta um número máximo de centrais e um custo por central.

6. Conclusão

6.1. Dificuldades

Numa fase inicial, tivemos alguma dificuldade a interpretar o enunciado e a perceber como seriam introduzidos os dados. A principal dificuldade sentida foi na segunda tarefa, uma vez que foram precisas várias modificações ao algoritmo de Kruskal de modo a obter o resultado pretendido.

6.2. Contribuições

- Miguel Pereia – 70%
- Luís Oliveira – 15%
- Francisco Rodrigues – 15%