Respostas do Aquecimento para a Prova 1 (P1) – Grafos (INE5413)

Ciências da Computação – Universidade Federal de Santa Catarina

Prof. Rafael de Santiago

Nome:
Matrícula:

Observações gerais:

- A prova deverá ser concluída até as 10h00m;
- Pode ser utilizado material para consulta;
- Não será permitido compartilhamento de material de consulta.

1. *(2.5pt)* Considere o seguinte problema: Em um setor de investigações da polícia em uma determinada cidade, têm-se acesso às seguintes informações de ligações telefônicas que acontecem na área: dados sobre os contatos telefônicos (nome e CPF do proprietário da linha); listagem de quem ligou para quem dentro da mesma cidade; e duração de cada ligação. Recentemente, há uma demanda no setor relacionada a identificar as pessoas próximas a um indivíduo sendo investigado. No contexto de ligação telefônica, os investigadores consideram apenas ligações com mais de 30 segundos. São consideradas pessoas próximas, a distância de até duas ligações. Ou seja, se o indivíduo A ligou para B que ligou para C e as ligações duraram mais de 30 segundos cada, considera-se que A, B e C são indivíduos "próximos" para o contexto investigativo, inclusive se A e C não tiverem registros de ligações com mais de 30 segundos entre si. Com base nesse problema, responda:

   (a) *(0.5pt)* Qual tipo de grafo poderia ser utilizado para o problema (dirigido ou não, ponderado ou não)? Justifique.

   Resposta: *Para representar o problema, se utilizaria um grafo não-dirigido e não-ponderado. Não-ponderado porque é importante tomar nota se houve ou não ligações entre dois indivíduos. Não-dirigido porque a origem e o destino da ligação é ignorada, já que se considera que os dois envolvidos são considerados próximos apenas por entrarem em contato.*

   (b) *(1.0pt)* Como seria a montagem do grafo de entrada para o problema?

   Resposta: *Cria-se um vértice $G = (V, E)$, onde V é o conjunto de indivíduos e E é o conjunto de ligações, no qual cada aresta $\{u, v\} \in E$ representa uma ligação telefônica que durou mais de 30s entre os indivíduos u e v.*

   (c) *(1.0pt)* Qual dos algoritmos estudados poderia ser utilizado? O que deveria ser modificado nesse algoritmo?

Resposta: *Devido a propriedade de caminho mínimo para grafos não-ponderados (Teorema 3.1.5 nas Anotações da Disciplina), encontra-se a conexão entre cada par de indivíduos com o menor número de arestas através de uma busca em largura. Então, utilizaria uma busca em largura modificada. Essa busca não incluiria vizinhos de vértices com $d_u \geq 2$ na fila.*

2. *(2.5pt)* Considere o seguinte problema: Uma empresa quer participar de um edital para conectar várias localizações utilizando uma linha férrea. No levantamento, há 1000 localidades denominadas de $l_1, l_2, \ldots, l_{1000}$, nas quais l_1 é o começo e o destino da linha férrea. Uma das questões principais para participar da licitação é ser competitivo no preço da grande obra. Nesse contexto, a empresa deseja encontrar um ciclo que atenda as localidades sem a necessidade de construir pontes, atravessar morros e realizar grandes desapropriações de terra. Para isso, os técnicos da empresa se reuniram e definiram uma listagem de pares de localidades que não poderiam ser conectadas pois inviabilizariam o projeto. Deseja-se então desenvolver um programa de computador que faça o cáculo se é viável ainda conectar todos os pontos através de um ciclo. Com base no relatado acima, responda:

   (a) *(1.25pt)* Como seria a montagem do grafo de entrada para o problema?

   Resposta: *Um grafo $G = (V, E)$ não-dirigido e não-ponderado seria criado, no qual V seria o conjunto de localidades. A criação do conjunto de arestas se daria em dois passos: (i) primeiro, consideraria-se que cada par de vértices são conectados (grafo completo), então $E = V \times V$; (ii) depois, removeria-se as arestas correspondentes à listagem de pares de localidades que não poderia ser conectadas.*

   (b) *(1.25pt)* Qual dos algoritmos estudados poderia ser utilizado? O que deveria ser modificado nesse algoritmo?

   Resposta: *Como se necessita de um ciclo que passe por todas as localidades e dadas as restrições de custo, considera-se encontrar um ciclo hamiltoniano. Portanto, utilizaria-se o algoritmo de Bellman-Held-Karp para procurá-lo.*

3. *(2.5pt)* Sabe-se que o algoritmo de Floyd-Warshall tem complexidade de tempo $\Theta(|V|^3)$ para encontrar os caminhos de menor custo entre todos os vértices de um grafo $G = (V, E, w : E \to \mathbb{R})$. Considerando que G tenha $|E| \approx |V|$ e não tenha arestas com pesos negativos, faça:

   (a) *(1.5pt)* Crie um algoritmo mais eficiente que Floyd-Warshall para encontrar os caminhos mínimos para cada par de vértices em G. Se for necessário, considere que está a disposição os algoritmos de Bellman-Ford (complexidade de tempo $O(|V||E|)$) e Dijkstra (complexidade de tempo $O(|V| + |E|) \log_2 |V|$)

   Resposta: *O algoritmo para atender a solicitação se encontra abaixo. Nele, descobre-se o custo dos menores caminhos a partir de cada vértice para cada outro em G.*

   Algorithm 1: Algoritmo da Questão 3(a).

   Input : um grafo ponderado $G = (V, A, w)$
   1 criar matriz de distâncias D com $\mathbb{R}^{|V|\times|V|}$
   2 preencher matriz com $d_{uv} \leftarrow \infty$ para todo $u, v \in V$
   3 **foreach** $v \in V$ **do**
   4     $(D', A') \leftarrow$ Dijkstra(G, v)
   5   **foreach** $u \in V$ **do**
   6         $d_{vu} \leftarrow D'_u$
   7 **return** D

   (b) *(0.5pt)* Calcule a complexidade de tempo do algoritmo produzido;

   Resposta: *A definição inicial da matriz nas linhas 1 e 2 consome $\Theta(|V|^2)$ operações. O laço da linha 3 realiza $|V|$ vezes a chamada do algoritmo de Dijkstra ($O(|V|+|E|) \log_2 |V|$) e*

$|V|$ vezes um laço de repetição que passa sobre cada vértice. Como $|E| \approx |V|$, cada chamada do Dijkstra demandaria $O(|V| \log_2 |V|)$. Então, têm-se a complexidade de $O(|V|^2 \log_2 |V|)$ que é dominada assintoticamente por $\Theta(|V|^3)$, ou seja, o algoritmo proposto é mais eficiente que Floyd-Warshall.

(c) *(0.5pt)* Caso não tenha usado um dos algoritmos dispostos (Bellman-Ford ou Dijkstra), justifique.

Resposta: *O uso de Bellman-Ford se justificaria apenas se o grafo de entrada possuísse arestas de peso negativo, pois Dijkstra é mais eficiente que Bellamn-Ford para grafos com pesos não-negativos.*

4. *(2.5pt)* Seja $G = (V, A, w)$ um grafo dirigido e ponderado, crie um algoritmo para determinar os caminhos mínimos a partir de todo $v \in V$ para um vértice t. Caso seja necessário, pode-se chamar os algoritmos Bellman-Ford (complexidade de tempo $O(|V||E|)$) e Dijkstra (complexidade de tempo $O(|V| + |E|) \log_2 |V|)$ como se já estivessem implementados.

Resposta: *O algoritmo para atender a solicitação se encontra abaixo. No algoritmo, por inverter o sentido dos arcos, ao executar um algoritmo como Bellman-Ford ou Dijkstra com t como vértice de "origem" (agora destino), se obtém os caminhos mínimos de todos os vértices para t.*

Algorithm 2: Algoritmo da Questão 4.

Input : um grafo dirigido e ponderado $G = (V, A, w)$, um vértice de destino t

$A' \leftarrow \{\}$
criar função $w' : A' \rightarrow \mathbb{R}$
foreach $(u, v) \in A$ **do**
  $A' \leftarrow A' \cup \{(v, u)\}$
  define $w'\big((v, u)\big) \rightarrow w\big((u, v)\big)$
$G' \leftarrow (V, A', w')$
`/*` D_u `define as distâncias a partir de` u `até` t `e` S_u `seria o sucessor de` u `no caminho até` t`.` `*/`
$(D, S) \leftarrow$ Bellman-Ford(G', t)
return (D, S)

Boa Prova!