

PUC-Rio
Departamento de Informática
Profs. Marcus Vinicius S. Poggi de Aragão
Período: 2002.2
12 de outubro de 2002
Horário: 2as-feiras e 4as-feiras de 16 às 18 horas

ANÁLISE DE ALGORITMOS (INF 1721)

2ª Lista de Exercícios

1. Dado um grafo não-orientado $G = (V, E)$ e pesos $w_e, e \in E$, associados às arestas, proponha algoritmos, com suas respectivas estruturas de dados, para as seguintes operações:
 - (a) Dado um conjunto $X \subset V$, encontrar o valor do corte definido por X : $\sum_{(i,j)|i \in X, j \in V-X} w_{(i,j)}$. Isto é, a soma dos pesos associados às arestas com uma extremidade em X e outra em $V - X$.
 - (b) Calcular o novo valor de um corte quando um elemento, j , é retirado do conjunto X . A complexidade deve ser $O(n^2)$. Explique por que essa operação é $\Omega(n^2)$.
 - (c) Obter o elemento (vértice) cuja retirada do conjunto X impõe a maior redução sobre o valor do corte. Qual a complexidade de se obter a variação no valor do corte para a saída de cada vértice.
 - (d) Dado que as variações estão disponíveis, antes da retirada de um dado vértice, proponha um algoritmo $O(n)$ para determinar as variações (para todos os vértices) após a retirada do vértice dado.
 - (e) Proponha um algoritmo para fazer a retirada sucessiva de todos elementos em X , a cada iteração retira-se sempre o que corresponde à maior redução (ou menor aumento, caso não seja possível uma redução). O algoritmo deve executar em $O(n^2)$.
 - (f) Repita os itens acima para a inclusão dos elementos que não estão em X .
 - (g) Repita agora os itens acima para a inclusão dos elementos que não estão em X e a exclusão dos que não estão.
2. Apresente um algoritmo para transformar o problema (MC) no problema (MS) em tempo linear.

Clique-Máximo (MC) - Dado um grafo não-orientado $G = (V, A)$ e uma constante K . Pergunta-se se este grafo G possui um clique (isto é um sub-grafo completo) de cardinalidade maior ou igual à K .

Estável-Máximo (MS) - Dado um grafo não-orientado $G = (V, A)$ e uma constante K . Pergunta-se se este grafo G possui um conjunto de vértices independentes (isto é, um conjunto de vértices onde não existe aresta entre nenhum par do conjunto) de cardinalidade maior ou igual à K .
3. Apresente um algoritmo para transformar o problema (PART) no problema (KP) em tempo linear.

(PART) Instância: Dado um conjunto de elementos $X = \{x_1, x_2, \dots, x_n\}$ e seus respectivos tamanhos $T = \{t_1, t_2, \dots, t_n\}$.

Questão: Existe um subconjunto S de X tal que $\sum_{x_i \in S} T_i = \sum_{x_i \in X-S} T_i$?

(KP) Instância: Dado um conjunto de elementos $X = \{x_1, x_2, \dots, x_n\}$, seus respectivos tamanhos $T = \{t_1, t_2, \dots, t_n\}$ e seus respectivos valores $V = \{v_1, v_2, \dots, v_n\}$, e as constantes T_0 e V_0 .

Questão: Existe um subconjunto de S de X tal que a soma dos tamanhos dos elementos em S seja menor ou igual a T_0 e tal que a soma dos seus respectivos valores seja superior ou igual a V_0 ?

4. Considere o algoritmo de Dijkstra para encontrar o caminho mais curto entre um vértice *fonte* e os demais vértices de um grafo orientado $G = (V, E)$ onde a distância associada a um arco $e \in E$ é dada por $l_e = l_{vw}$ onde v e w são o vértice de partida e de chegada do arco e , respectivamente. (As complexidades devem ser obtidas em função de $n = |V|$ e $m = |E|$).

Algoritmo Dijkstra (s - fonte)

Passo 0: *Inicialização*

Seja S o conjunto de vértices com caminho mais curto a partir de s determinado, e \bar{S} seu complemento ($\bar{S} = V - S$).

$S \leftarrow \emptyset$

$d(i) \leftarrow +\infty \forall i \in V$;

$d(s) \leftarrow 0$; $pred(s) \leftarrow 0$;

Passo 1: *Iteração*

Enquanto $S \subset V$ faça

1.1 Encontre $v \in \bar{S}$ t.q. $d(v) = \min_{w \in \bar{S}} d(w)$

1.2 $S \leftarrow S \cup \{v\}$; $\bar{S} \leftarrow \bar{S} \setminus \{v\}$;

1.3 Para todo $w \in \Gamma^+(v)$

Se $d(v) + l_{vw} < d(w)$

então $d(w) \leftarrow d(v) + l_{vw}$; $pred(w) \leftarrow v$;

Responda aos itens abaixo:

- (a) Considere que o passo **1.1** é realizado através do uso de um vetor que armazena os valores $d(i)$, $i = 1, \dots, n$. Qual a complexidade deste passo (uma execução) ? Qual a complexidade total (soma de todas as execuções) deste mesmo passo (1.1) ?
- (b) Repita esta análise para os passos (1.2) e (1.3).
- (c) Qual a complexidade global desta implementação do algoritmo de Dijkstra ?
- (d) Onde poderia ser usada uma *heap* ? Por que isso poderia ser bom ?
5. Sejam P_1 , P_2 e P_3 três problemas tais que $P_1 \leq_n P_2 \leq_{n^3 \log n} P_3$ (i.e., P_1 é redutível a P_2 em tempo linear e P_2 a P_3 em tempo $n^3 \log n$). Assuma a hipótese de que P_1 é $\Omega(n \log n)$. Assuma também que você conhece um algoritmo $O(n^3)$ para resolver P_3 .

Discuta as afirmações abaixo.

- (a) O que você pode dizer sobre a complexidade de resolução de P_2 ? Qual a complexidade do melhor algoritmo que você conhece para P_2 ?
- (b) Todo algoritmo que resolve P_2 tem que gastar pelo menos tempo quadrático (P_2 é $\Omega(n^2)$).
- (c) $\Omega(n \log n)$ é um limite inferior para a complexidade de P_3 .
- (d) P_2 pode ser resolvido no pior caso em tempo $O(n \log n)$.
6. Considere uma árvore $T = (V, E)$ com distâncias d_e associadas às arestas. Deseja-se construir uma matriz $n \times n$ contendo as distâncias entre cada par de vértices sobre T . Considere que T é dada no formato de lista de adjacência. Deseja-se um algoritmo $O(n^2)$ para se construir a matriz de distâncias.
7. Suponha que $2n$ alunos querem entrar em n universidades. Cada universidade vai receber no máximo 2 alunos e você conhece a lista de universidades que aceitaram cada aluno. Deseja-se alocar os alunos às universidades de modo que o máximo de alunos ingressem nas universidades. Formule esse problema como um problema de **Alocação/Emparelhamento** em um grafo bipartido. Lembre que você deve especificar uma instância do problema de Alocação que resolve esse problema, isto é, definir a matriz de custos de atribuição.
8. Considere o seguinte problema: (CMG) Caminho de menor Gargalo: Dado um grafo não-orientado $G = (V, E)$ com distâncias d_e associadas às arestas, e um vértice r , determinar os caminhos de r a todos os demais vértices, cuja maior aresta é o menor possível.
- Transforme o problema de encontrar a Árvore Geradora Mínima em grafo no problema (CMG).
9. Considere um grafo não-orientado $G = (V, E)$ com pesos w_e associados às arestas. Um conjunto $F \subseteq E$ é chamado de **conjunto de arestas de realimentação** se todos os ciclos em G contém pelo menos uma aresta de F . Deseja-se encontrar o conjunto F de menor peso.
- Transforme este problema no de encontrar a árvore geradora de peso **máximo** (dê um algoritmo para esse problema).
- Dica:** Observe que o grafo obtido retirando as arestas em F de $G = (V, E)$ não pode conter ciclos (por que ?).