

PUC-Rio
Departamento de Informática
Profs. Marcus Vinicius S. Poggi de Aragão
Período: 2007.2
Horário: 2as-feiras e 4as-feiras de 19-21
21 de novembro de 2007

ESTRUTURAS DISCRETAS (INF 1308)

3ª Lista de Exercícios

Procure ser conciso e preciso nas suas argumentações.

1. Um comerciante possui um armazém que utiliza para suprir seus clientes de um único produto. O seu armazém pode guardar até C unidades do produto. Para as próximas T semanas o comerciante TEM que atender às demandas dos seus clientes que somam d_t para a semana t , onde $t = 1, 2, \dots, T$. Além disso, ele possui $s_0 (\leq C)$ unidades em estoque antes do início da primeira semana, e já negociou com os fornecedores os preços unitários p_t ($t = 1, 2, \dots, T$). Ele deseja planejar o atendimento dos seus clientes de modo a gastar o mínimo possível com a compra do produto.

Ajude ao comerciante a definir a sua estratégia ótima de compra do produto nas semanas $t = 1, \dots, T$.

- (a) Apresente o algoritmo que obtém a estratégia de compra de menor custo e atende às demandas dos seus clientes.
 - (b) Explique que reforço da hipótese indutiva deve ser utilizado para se obter um algoritmo eficiente para este cálculo.
 - (c) Mostre como fica a prova do passo indutivo e sua relação com o algoritmo proposto.
 - (d) Execute o seu algoritmo sobre a seguinte instância: $C = 12$, $T = 5$, $s_0 = 3$, $d_1 = 7$, $d_2 = 4$, $d_3 = 15$, $d_4 = 10$, $d_5 = 7$ e $p_1 = 3$, $p_2 = 4$, $p_3 = 7$, $p_4 = 6$, $p_5 = 8$. Informe quanto o comerciante deve comprar em cada semana e o seu custo total.
2. Seja uma sequência de n inteiros distintos $T = \{t_1, \dots, t_n\}$.
 - (a) Utilize indução matemática (com reforço de hipótese) para propor um algoritmo que obtenha a maior subsequência crescente (não necessariamente consecutiva).
 - (b) Mostre como os algoritmos obtidos acima funcionam na sequência:

3, 17, 9, 12, 35, 6, 27, 8, 21, 26, 23, 11, 19, 13, 15

3. Prove por indução e dê o algoritmo recursivo que resulta da sua prova.

Teorema: Todos os vértices de uma qualquer árvore podem ser coloridos (sem ter vértices adjacentes com uma mesma cor) com duas cores. Uma árvore é um grafo não orientado $G = (V, E)$ sem ciclos, conexo, e onde $|E| = |V| - 1$.

Dicas: use indução simples nos vértices e observe que qualquer árvore pode ser construída a partir de um vértice sozinho, com a inserção sucessiva de um vértice conectado a uma aresta.

Você deve escrever o algoritmo para colorir uma árvore qualquer com duas cores.

4. Seja $G = (V, E)$ um grafo **orientado** e acíclico, então existe uma renumeração dos seus vértices tal que todos os vértices que podem ser atingidos a partir de um vértice v , $v \in V$, estão renumerados com valores superiores a v .

Prove por indução o teorema abaixo e dê o algoritmo recursivo que resulta da sua prova.

Teorema: Todo grafo conexo e acíclico possui pelo menos um vértice com grau de entrada (número de arcos que chegam em um vértice) igual a ZERO.

Dicas: use indução simples nos vértices. Utilize $n = 2$ como caso base e mostre que o teorema vale para esta caso por exaustão.

Você deve escrever um algoritmo que encontre uma renumeração dos vértices que satisfaça a propriedade descrita no início da questão. A entrada do algoritmo é um grafo, i.e. um conjunto de vértices e um conjunto de arcos. A saída é a renumeração dos vértices, i.e. por exemplo: os vértices 1, 2, 3, 4 e 5 devem ser renumerados como 2, 4, 1, 5, 3.

5. Seja $G = (V, E)$ um grafo **não-orientado**. Deseja-se decidir se G é um grafo bi-partido ou não.

Um grafo é bi-partido se existe uma partição de V , V_1 e V_2 ($V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$) tal que $\forall (u, v) \in E$, tem-se que $u \in V_1$ e $v \in V_2$ ou vice-versa. Isto é, não existe aresta entre vértices de V_1 ou V_2 .

Caso G seja bi-partido, deseja-se obter a partição V_1 e V_2 de V .

Elabore o caso base e a hipótese indutiva para a prova, por indução, de que sabe-se encontrar um tal conjunto. Descreva o algoritmo resultante de sua prova.

6. Logo após uma operação de sucesso dois ladrões decidem se separar. Entretanto, o resultado da operação foi um conjunto de objetos indivisíveis (de alta liquidez) cujos valores de mercado são bem estabelecidos e conhecidos pelos ladrões (de alto nível, é claro), o que dificulta a partilha. O problema dos ladrões é dividir entre eles o conjunto de objetos roubados de modo que a diferença entre o valor total que cada um vai ficar seja a menor possível. Sabendo que são n objetos e que seus valores são v_1, v_2, \dots, v_n , responda aos itens abaixo.

- (a) Utilizando indução matemática (com reforço de hipótese), mostre como pode-se construir uma divisão dos objetos que tenha diferença de valor a menor possível.
- (b) (1.0) Descreva, de forma compacta, o algoritmo resultante.

- (c) (1.0) Mostre como o seu algoritmo funciona caso sejam 5 objetos com os seguintes valores: $v_1 = 5, v_2 = 4, v_3 = 7, v_4 = 3, v_5 = 17$.

Dica: Reforce a hipótese de modo que as divisões dos objetos sejam conhecidas para cada valor total atribuído a cada ladrão.

Suponha que antes de chegar ao local da partilha os ladrões tiveram que carregar os objetos. Assuma que os pesos dos objetos são p_1, p_2, \dots, p_n e que os ladrões possuem juntos uma mochila que resiste ao peso máximo de P .

- (d) Utilize indução matemática (com reforço de hipótese) para projetar um algoritmo que encontra o maior valor total, em objetos, que os ladrões podem carregar na mochila (represente por $V(k)$ este valor para os objetos $1, 2, \dots, k$; reforce a hipótese assumindo que este valor pode ser calculado, dado que o peso total dos objetos na mochila é exatamente q , para todos os pesos possíveis; represente por $V(k, q)$ este valor).
- (e) Mostre como o seu algoritmo funciona caso sejam 5 objetos com os valores: $v_1 = 5, v_2 = 4, v_3 = 7, v_4 = 3, v_5 = 10$ e os respectivos pesos $p_1 = 5, p_2 = 3, p_3 = 6, p_4 = 4, p_5 = 7$ onde o peso máximo suportado pela mochila é 9.
- (f) Como você resolveria esse problema se os ladrões tivessem duas mochilas ?

7. Sejam duas seqüências de inteiros distintos $T = \{t_1, \dots, t_n\}$ e $Q = \{q_1, \dots, q_m\}$.

- (a) Utilize indução matemática (com reforço de hipótese) para propor um algoritmo que obtenha a maior subsequência (não necessariamente consecutiva) comum às duas seqüências (T e Q).
- Qual é o reforço de hipótese utilizado ?
 - Explique a recursão resultante (i.e., o algoritmo):
- $$custo(i, j) = \max\{custo(i-1, j), custo(i, j-1), custo(i-1, j-1) + \delta\}$$
- onde δ vale 1 se $t_i = q_j$ e 0 caso contrário.
- Qual o valor dos custos: $custo(0, 0), custo(i, 0)$ para $i = 1, \dots, n$ e $custo(0, j)$ para $j = 1, \dots, m$?
- (b) Considere que para cada par de elementos de T e Q , t_i e q_j é associado um custo positivo c_{ij} , $i = 1, \dots, n, j = 1, \dots, m$. Proponha um algoritmo para encontrar a subsequência comum onde a soma dos custos dos elementos é maximizada.
- O que muda na recursão resultante ? Apresente a nova recursão resultante.
- (c) Mostre como os algoritmos obtidos acima funcionam nas seqüências:

3, 7, 9, 2, 3, 6, 2, 8

e

3, 5, 8, 7, 9, 2, 3, 6, 7

Execute o algoritmo do item (b) com os custos $c_{ij} = i + j$.

8. Considere o problema de encontrar a sub-árvore de uma árvore enraizada no vértice r cujo peso é máximo. A sub-árvore também deverá ser enraizada em r . O problema é definido da seguinte forma: dada uma árvore (é um grafo não-orientado, é claro) $T = (V, E)$, um vértice raiz $r \in V$ e pesos w_v associados aos vértices de V , determinar uma árvore $T^* = (V^*, E^*)$, contida em T onde $r \in V^*$ e a soma $\sum_{v \in V^*} w_v$ é máxima.

A árvore vazia vale zero(0) e, naturalmente, os valores dos pesos podem ser positivos ou negativos. Responda:

- (a) Use reforço de hipótese para propor um algoritmo que obtenha uma sub-árvore de peso máximo (nesse caso, o reforço é supor que conhece-se a sub-árvore de peso máximo com raiz em cada um dos vértices filhos, e indução prova que se é verdade para os filhos então é verdade para os pais).

Represente a árvore determinando para cada vértice o seu vértice pai ($p(v)$ é o pai de v).

- (b) Execute o seu algoritmo sobre a seguinte instância: $r = 1$, $V = \{1, \dots, 12\}$, $E = \{(1, 2), (1, 3), (2, 4), (2, 5), (2, 6), (3, 7), (3, 8), (5, 9), (5, 10), (8, 11), (8, 12)\}$, $w_1 = -2$, $w_2 = -1$, $w_3 = -1$, $w_4 = 4$, $w_5 = -6$, $w_6 = -1$, $w_7 = -1$, $w_8 = 1$, $w_9 = 2$, $w_{10} = 3$, $w_{11} = -2$ e $w_{12} = 3$.