PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2006.2
Horário: 2as-feiras e 4as-feiras de 9 às 11 horas - Sala 422L

ANÁLISE DE ALGORITMOS (INF 1721)

Lista 3

1. Sejam P_1, P_2 e P_3 três problemas tais que $P_1\ \alpha_n\ P_2\ \alpha_{n^3\, log\, n}\ P_3$ (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 $O(n\, log\, n)$.

2. Defina as classes de problemas P, NP e $NP - completo$ Relacione estas classes e dê um exemplo de problema para cada classe.

3. Seja P o conjunto dos problemas para os quais existem algoritmos determinísticos polinomias para a sua resolução . Seja NP o conjunto dos problemas para os quais existem algoritmos **não**-determinísticos polinomias para a sua resolução . Naturalmente P está contido em NP. Considere os problemas $P_1 \in P$ e $P_2 \in NP - completo$. Indique se cada afirmação abaixo é verdadeira, falsa ou se não se sabe.
   (a) Conhece-se uma redução de P_1 para P_2 que toma tempo polinomial ($O(n^k)$).
   (b) Se existe um algoritmo determinístico polinomial para a resolução de P_2 então podemos afirmar que $P_1 \in NP - completo$ assim como $P_2 \in P$.
   (c) P_2 é pelo menos tão difícil quanto $3 - SAT$.
   (d) $3 - SAT$ é pelo menos tão difícil quanto P_2.
   (e) Conhece-se uma redução de P_2 para P_1 que toma tempo polinomial.

4. Prove que os problemas (de decisão) abaixo pertencem à NP.
   (a) Árvore Geradora Mínima com restrição de Grau (AGG) - Dado um grafo não-orientado $G = (V, E)$, pesos $w_e \in E$ e constantes K e C.
   Pergunta-se se este grafo G possui uma árvore geradora cujo grau em nenhum dos vértices seja superior a K e cuja soma dos pesos das arestas nesta árvore seja no máximo C.
   (minimize C)
   (b) 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.
   (maximize K)

(c) (MS): Dados um conjunto de máquinas $M = \{m_1, m_2, \ldots, m_p\}$ e um conjunto de tarefas $T = \{t_1, t_2, \ldots, t_q\}$ cada uma com uma duração $d_i \in Z$, $i = 1, \ldots, q$ onde d_i associada e uma constante K. Considerando-se que as tarefas podem ser atribuídas à qualquer máquina indistintamente. Pergunta-se se existe uma atribuição das tarefas às p máquinas tal que o instante em que a última tarefa é terminada é menor ou igual que K (Isto é, a duração total é inferior ou igual a K).

(minimize K)

5. Considere a multiplicação de n matrizes $A_1, \ldots, A_n$. Considere também que denota-se o produto das matrizes $A_k.A_{k+1}.\ldots.A_q$ por $A_{k..q}$ e que as dimensões das matrizes são dadas por $d_k \times d_{k+1}$ para a matriz A_k. Assim, $A_{1..n}$ terá dimensão $d_1 \times d_{n+1}$.

   Aqui a multiplicação de pares consecutivos de matrizes $A_k.A_{k+1}$ é feita calculando

   $$a_{i,j}^{k..k+1} = \sum_{p=1}^{d_{k+1}} a_{i,p}^{k}.a_{p,j}^{k+1}$$

   para todo par (i, j) que é elemento de $A_k.A_{k+1}$ e onde $a_{i,j}^k$ representa o elemento (i, j) da matriz A_k.

   Observe que a multiplicação de 3 matrizes A_1, A_2 e A_3, pode ser feita de duas manieras: $((A_1.A_2).A_3)$ e $(A_1.(A_2.A_3))$. (De quantas maneiras pode-se obter o produto de n matrizes ?)

   Observe também que para cada maneira de se multiplicar n matrizes pode-se ter que realizar um número diferente de multiplicações. Quantas são ?

   Apresente um algoritmo para determinar a maneira de se multiplicar as n matrizes que utiliza o menor número de multiplicações. Analise a complexidade do algoritmo proposto. A complexidade é polinomial ? Qual a complexidade menor possível que um algoritmo que resolve este problema pode ter ?

6. 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 $t = 1, 2, \ldots, T$. 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 $(t = 1, 2, \ldots, T)$. 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 $t = 1, \ldots, T$.

   (a) Apresente o algoritmo que obtém a estratégia de compra de menor custo e atende às demandas dos seus clientes.

   (b) Analise a complexidade do algoritmo proposto. A complexidade é polinomial ? Qual a complexidade menor possível que um algoritmo que resolve este problema pode ter ?

   (c) 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.

7. Considere um tabuleiro de xadrez e um rei que está inicialmente na posição $(1, 1)$ (as posições do tabuleiro são representadas por (i, j) onde $1 \leq i \leq 8$ e $1 \leq j \leq 8$). Para cada posição do tabuleiro estão associados um prêmio p_{ij} e um consumo q_{ij} (o prêmio pode ser em USD(!!) e o consumo em litros de gasolina, por exemplo). Os prêmios e os consumos assumem somente valores positivos. O rei tem inicialmente Q unidades para consumir e pode passar quantas vezes quiser em cada posição do tabuleiro e a cada vez receber o prêmio e, naturalmente, consumir os seus recursos. Ao final (do passeio) **o rei tem que estar de volta na posição** $(1, 1)$.

(a) Proponha um algoritmo para determinar o caminho que o rei deve fazer para obter o maior total possível em prêmios.

(Dica: Suponha que você conhece a solução que obtém o maior total em prêmios dado que o rei está em cada uma das posições do tabuleiro e para cada consumo possível. Escreva agora o teorema do passo indutivo reforçando a hipótese indutiva. A prova por indução matemática (simples) de que você sabe resolver o problema do rei leva ao algoritmo).

(b) Analise a complexidade do algoritmo proposto. A complexidade é polinomial ? Qual a complexidade menor possível que um algoritmo que resolve este problema pode ter ? Qual a complexidade deste problema ?

(c) Suponha que $Q = 7$ e que $q_{ij} = 1$ e $p_{ij} = 2 * i + 3 * j$ para todo (i, j). Determine o caminho em que o rei acumula o maior total possível de prêmios. Repita o cálculo modificando apenas $p_{22} = 100$ e mantendo os demais valores.

8. Sejam $T = \{t_1, \ldots, t_n\}$ e $P = \{p_1, \ldots, p_k\}$ duas sequências de caracteres on $k \leq n$.

(a) Proponha um algoritmo linear para determinar se P é uma subsequência de T, isto é, se os elementos de P aparecem em T na mesma ordem que em P, mas não necessariamente consecutivos.

(b) Suponha que a resposta do item anterior é negativa. Proponha um algoritmo para encontrar a maior subsequência de P que está em T.

(c) Considere que para cada elemento de T é associado um custo positivo c_i, $i = 1, \ldots, n$. Proponha um algoritmo para encontrar a subsequência de P que é subsequência de T onde a soma dos custos dos elementos de T é maximizada.

(d) Analise a complexidade dos algoritmos propostos. Qual a complexidade menor possível de um algoritmo que resolve estes problemas ?