PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2004.2
Horário: 3as-feiras e 5as-feiras de 13 às 15 horas - Sala 422L

PROJETO E ANÁLISE DE ALGORITMOS (INF 2926)

Lista 1

1. Considere as seguintes funções:

   (a) $10.n^{\pi}$
   (b) $log\, n$
   (c) $log\, (n^2)$
   (d) $0.005.n^{0.00001}$
   (e) $1000.(log\, n)^2$
   (f) $30.n^3.log\, n$
   (g) $50.n.log^2\, n$
   (h) $(log\, n)^{log\, n}$
   (i) $\frac{n}{log\, n}$
   (j) $70.n$
   (k) $log\, log\, n$
   (l) $(1.5)^{(log\, n)^2}$
   (m) $90.n^2.log\, n\ + n^3.log\, n$

   - Coloque as funções acima em ordem de crescimento assintótico, i.e. valor quando $n \longrightarrow \infty$.
   - Utilize pelo menos três vezes cada um dos símbolos O, Ω, Θ, o, e ω para indicar a relação existente entre pares das funções acima (não vale recíprocos).

2. Verdadeiro ou Falso. Justifique.

   (a) $(log\, n)^{100} = O(n^{\epsilon}),\ \forall \epsilon > 0.$
   (b) $2^{n+1} = O(2^n).$
   (c) Se $g(n,m) = mlog_d n$ onde $d = [m/n+2]$ ($[x]$ é o menor inteiro maior que x), $m = O(n^2)$, então $g(n,m) = O(m^{(1+\epsilon)})\ \forall \epsilon > 0.$

3. Encontre contra-exemplos para as proposições abaixo.

   (a) Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) - g(n) = O(s(n) - r(n))$
   (b) Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n)/g(n) = O(s(n)/r(n))$

4. 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, **considerando que somente o conhecimento acima está disponível**.
   (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) Todo algoritmo que resolve P_1 pode ser usado para resolver P_2.
   (e) Todo algoritmo que resolve P_3 pode ser usado para resolver P_2.
   (f) P_2 pode ser resolvido no pior caso em tempo $O(n \, log \, n)$.

5. 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) Existe uma redução de P_2 para P_1 que toma tempo polinomial.

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

7. Dado que você conhece um algoritmo determinístico polinomial para a resolução do problema (CMC) abaixo, USE ESTE CONHECIMENTO para provar que você também conhece um algoritmo determinístico polinomial para a resolução do problema (VMC) apresentado em seguida.

   (CMC) Dado um grafo orientado $G = (V, A)$ com comprimentos positivos associados aos arcos $(i, j) \in A$, dois vértices $i, j \in V$ e uma constante K. Pergunta-se se existe um caminho do vértice i ao vértice j que possua comprimento menor ou igual à K.

   (VMC) Dado um grafo orientado $G = (V, A)$, com comprimentos positivos associados aos arcos $(i, j) \in A$, e uma constante K. Pergunta-se se existe um par de vértices (i, j) para o qual exista um caminho de i a j e de j a i que tenha comprimento menor ou igual à K.