MC538/MC438: Análise de Algoritmos II

Turmas A/B – Profs. Cid C. de Souza e Zanoni Dias

Lista 1

15 de março de 2005

1. Sejam P_1 e P_2 dois problemas tais que $P_1 \propto_n P_2$ e suponha que P_1 tem cota inferior $\Omega(n \log n)$, onde n é um parâmetro que mede o tamanho da entrada do problema P_1. Quais das seguintes afirmações são verdadeiras ? Justifique cuidadosamente as suas respostas.
   (a) $\Omega(n \log n)$ também é cota inferior para P_2.
   (b) Todo algoritmo que resolve P_1 também pode ser usado para resolver P_2.
   (c) Todo algoritmo que resolve P_2 também pode ser usado para resolver P_1.
   (d) O problema P_2 pode ser resolvido no pior caso em tempo $O(n \log n)$.

2. Sejam P_1 e P_2 dois problemas tais que um deles tenha cota inferior $\Omega(n^k)$, para algum $k > 1$, num modelo computacional $\mathcal{M}$ e o outro é solúvel em tempo $O(n \log n)$ no mesmo modelo computacional $\mathcal{M}$. Se P_1 é redutível a P_2 em tempo linear, decida qual é qual. O parâmetro n denota o tamanho da entrada dos dois problemas.

3. Diz-se que um ponto $p = (x_p, y_p)$ do plano **domina** um outro ponto distinto $q = (x_q, y_q)$ do plano se $x_p \geq x_q$ e $y_p \geq y_q$. Um ponto p é dito ser **maximal** em relação a um conjunto de pontos S se $p \in S$ e nenhum ponto de S domina p.

   Projete um algoritmo de complexidade $O(n \log n)$ para encontrar todos os pontos maximais de um conjunto P contendo n pontos distintos no plano.

4. Considere o seguinte problema: dados n intervalos na reta real, definidos pelos seus pontos de início e de fim, projete um algoritmo que lista todos os intervalos que estão contidos dentro de pelo menos um dos outros intervalos passados na entrada. O seu algoritmo deve ter complexidade $O(n \log n)$.

5. Denote por `MAX` o problema do item 3 e por `INTERVAL` o problema do item 4. Encontre uma redução de complexidade linear de `MAX` para `INTERVAL`.

   É possível usar o algoritmo desenvolvido no item anterior e a redução proposta por você para projetar um algoritmo para `MAX` ? Em caso afirmativo, como se compara a complexidade deste algoritmo com àquela do algoritmo do item 3 ?

6. Encontre uma redução de complexidade linear de `INTERVAL` para `MAX`.

7. Usando o conceito de dominância entre pontos do item 3, pode-se definir os **Pareto**s de um dado conjunto não vazio de pontos $P = \{p_1, \ldots, p_n\}$ no plano da seguinte forma:
   (i) o **Pareto** 1 de P, denotado por P_1, é o conjunto de pontos maximais de P;
   (ii) para $i \geq 2$, o **Pareto** i de P, denotado por P_i, é o conjunto de pontos maximais de $P \setminus (P_1 \cup \ldots \cup P_{i-1})$.

Chamemos de **índice de Pareto** de P o menor valor de i para o qual o **Pareto** i é vazio[1]. Denotemos por $i(P)$ este valor.

Assim, dado um conjunto P como acima, considere o problema de encontrar $i(P)$ primeiros Paretos de P. Desenvolva um algoritmo $O(n \log n)$ para este problema.

8. Encontre uma redução polinomial do problema de ordenação de um vetor de n elementos para o problema `PARETO` do item anterior. A sua redução deve ter complexidade $O(n)$.

   Pergunta-se: esta redução prova que o algoritmo do item anterior é ótimo (do ponto de vista de complexidade computacional) ? Justifique a sua resposta.

9. Considere o seguinte problema: são dados um grafo direcionado $G = (V, E)$ com um vértice especial v e um custo $c(u) \geq 0$ para cada vértice u de V. Suponha que o custo de um caminho direcionado representado pela seqüência de vértices $\{v, x_1, x_2, \ldots, x_k, u\}$ seja dado por $\sum_{i=1}^{k} c(x_i)$, ou seja, o custo de um caminho é a soma do custo dos seus vértices internos. Assim, se (v, u) é um arco do grafo, o custo deste caminho é *zero*.

   Deseja-se encontrar um caminho menor custo entre v e todos os vértices de $V \setminus \{v\}$.

   Projete um algoritmo de complexidade polinomial para este problema usando uma redução que envolva o problema do caminho mais curto em grafos com custos nas arestas (veja por exemplo *U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley, 1989*, seção 7.5).

10. Uma matriz quadrada é dita ser **triangular inferior** (**superior**) se todos os seus elementos não nulos estiverem na diagonal principal ou abaixo (acima) dela.

    Seja `MMIS` o problema de multiplicar uma matriz triangular inferior por uma matriz triangular superior e `MMA` o problema de multiplicar duas matrizes quadradas arbitrárias.

    Seja $T(n)$ a complexidade de um algoritmo ótimo para resolver `MMIS` quando as matrizes passadas na entrada tem dimensão $n \times n$. Suponha que $T(cn) \in O(T(n))$ para toda constante $c > 0$.

    Mostre que `MMIS` é pelo menos tão difícil quanto `MMA` no sentido em que estes dois problemas tem a mesma cota inferior (supondo o modelo de computação usual).

11. Seja S um conjunto de n pontos distintos do plano. Seja $G = (V, E)$ o grafo não direcionado completo com n vértices de modo que exista uma relação $1 : 1$ entre os vértices de V e os pontos de S. Além disso, assuma que para cada aresta (u, v) em E, esteja associado um custo $c(u, v)$ que é igual à distância euclidiana entre os pontos correspondentes a u e v em S.

    Mostre que a cota inferior do problema de encontrar a árvore geradora mínima de G tem cota inferior $\Omega(n \log n)$.

    A definição do problema da árvore geradora mínima de um grafo pode ser encontrada na mesma referência citada no item 9 (veja seção 7.6).

12. Seja $G = (V, E)$ um grafo não direcionado tal que pra cada vértice v do grafo temos associado uma função $b(v) \leq grau(v)$. Um b-emparelhamento é um subconjunto de E tal que cada

[1] a partir deste ponto todos Paretos serão vazios !

vértice v não tem mais do que $b(v)$ arestas incidentes a ele. Um b-emparelhamento máximo é aquele que tem o maior número de arestas possível. Reduza o problema de se achar um b-emparelhamento máximo ao problema de se achar um emparelhamento máximo em um grafo.

Dica: Dado o grafo $G = (V, E)$, considere o seguinte grafo G'. Para cada $v \in V$ crie $b(v)$ vértices, $v_1, \ldots, v_{b(v)}$. Para cada aresta (u, v) de G crie os vértices e_{uv} e e_{vu} com aresta (e_{uv}, e_{vu}) ligando-os. Para cada vértice v_i crie aresta (v_i, e_{vy}), para cada y.