

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 2

1. Considere uma *heap* com  $n$  inteiros e um valor  $x$ . A *heap* possui o seu maior elemento no topo e está organizada em um vetor (de  $n$  posições). Proponha um algoritmo para decidir se o valor  $x$  é maior ou igual ao  $k$ -ésimo maior elemento na *heap* ou não. O algoritmo deve executar em  $O(k)$ .

Dica: Observe que você não precisa necessariamente encontrar o  $k$ -ésimo maior elemento da *heap*.

2. Considere o seguinte problema:

[ELE] Dados um conjunto de  $n$  inteiros distintos  $X = \{x_1, \dots, x_n\}$  e pesos positivos  $w(x_j)$ ,  $j = 1, \dots, n$ , associados aos elementos de  $X$ , e um inteiro positivo  $V$  tal que  $0 \leq V \leq W = \sum_{j=1}^n w(x_j)$ ; encontrar o elemento  $x_k$  tal que  $\sum_{x_j < x_k} w(x_j) < V$  e  $w(x_k) + \sum_{x_j < x_k} w(x_j) \geq V$ .

(a) Projete um algoritmo  $O(n \log n)$  para resolver esse problema.

(b) Utilize divisão e conquista para projetar um algoritmo  $O(n)$  para resolver esse problema.

(c) Apresente detalhadamente a análise da complexidade do item anterior.

3. Sejam  $u$  e  $v$  dois números de  $n$  bits (considere que  $n$  é potência de 2). A multiplicação tradicional de  $u$  por  $v$  utiliza  $O(n^2)$  operações. Um algoritmo baseado em divisão e conquista divide os números em duas partes iguais, calculando o produto como:  $uv = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd$ . As multiplicações  $ac$ ,  $ad$ ,  $bc$  e  $bd$  são feitas usando este mesmo algoritmo recursivamente.

- Escreva este algoritmo
- Determine a complexidade
- Qual é a complexidade se  $ad + bc$  é calculado como  $(a + b)(c + d) - ac - bd$ ?

4. Prove que os problemas abaixo pertencem a NP, ou explique porque não se sabe, ou porque não pertence a NP. Para provar que pertence à NP, apresente o certificado e justifique porque a verificação do SIM é feita em tempo polinomial.

HA Hemisfério Aberto

Instância: Um conjunto finito de  $n$   $m$ -tuplas  $X = \{x_1 = \langle x_1^1, x_1^2, \dots, x_1^m \rangle, x_2, \dots, x_n\}$  e um inteiro  $K \leq n$ .

Pergunta: Existe uma  $m$ -tupla  $y = \langle y^1, y^2, \dots, y^m \rangle$  tal que o produto interno entre  $y$  e  $x$  é maior que zero, isto é  $y \cdot x > 0$ , para pelo menos  $K$   $m$ -tuplas  $x \in X$  ?

KCM  $K$  Caminhos mais curtos.

Instância: Um grafo orientado  $G = (V, E)$ , as distâncias positivas dos arcos  $e$ ,  $\ell_e$ , para  $e \in E$ , um par de vértices de  $V$ ,  $s$  e  $t$ , e inteiros positivos  $L$  e  $K$ .

Pergunta: Existem  $K$  caminhos de  $s$  a  $t$  em  $G$  tais que sua distância total não ultrapassa  $L$  ?

EDQ Equações Diofantinas Quadráticas.

Instância: Inteiros positivos  $a$ ,  $b$  e  $c$ .

Pergunta: Existem inteiros positivos  $x$  e  $y$  tais que  $a.x^2 + b.y = c$  ?

STA Atribuição Sequencial V/F

Considere o seguinte jogo entre dois jogadores. Dada um instância do problema 3-SAT, os jogadores atribuem V ou F às variáveis em um sequência pré-fixada. Na primeira rodada o jogador 1 atribui V ou F à variável  $u_1$  e o jogador 2 à variável  $u_2$ , na segunda, 1 à  $u_3$  e 2 à  $u_4$ . Na  $i$ -ésima jogada, 1 à  $u_{2i-1}$  e o jogador 2 à  $u_{2i}$ . O jogador 1 vence se todas as cláusulas forem satisfeitas. Considere agora o seguinte problema:

Instância: Uma sequência de variáveis lógicas  $U = \{u_1, u_2, \dots, u_n\}$  e uma coleção de cláusulas de tamanho 3,  $C = \{c_1(u), c_2(u), \dots, c_n(u)\}$

Pergunta: O jogador 1 vence o jogo ? Ou seja, existe um sequência de decisões para o jogador 1 em que ele vence o jogo independente das ações do jogador 2 ?

5. Considere os seguintes problemas:

- *Vertex Cover* (VC): Dados um grafo (não orientado)  $G = (V, E)$  e um inteiro positivo  $K \leq |V|$ ; determinar se existe  $S \subset V$ ,  $|S| \leq K$ , tal que para toda aresta  $(u, v) \in E$ , pelo menos um entre  $u$  e  $v$ , pertence a  $S$ .
- *Dominating Set* (DS): Dados um grafo (não orientado)  $H = (N, A)$  e um inteiro positivo  $L \leq |N|$ ; determinar se existe  $U \subset N$ ,  $|U| \leq L$ , tal que para todo vértice  $w \in N \setminus U$  existe um vértice  $u \in U$  tal que  $(u, w) \in A$ .
- *Uncapacitated Facility Location* (UFL): Dados conjuntos  $M$  e  $N$ , inteiros  $c_{ij}$  para  $i \in M$  e  $j \in N$ , inteiros  $f_j$  para  $j \in N$  e um inteiro  $T$ ; determinar se existe um conjunto  $J \subseteq N$  tal que  $\sum_{i \in M} \min_{j \in J} c_{ij} + \sum_{j \in J} f_j \leq T$ .

(a) Sabendo que (VC) é NP-completo (foi provado em aula!), prove que (DS) também é NP-completo.

Dicas: Observe que todo VC é um DS. Pense em formas de obrigar que um DS também seja um VC, ou seja se cobrir todos os vértices no grafo do problema (DS) (o grafo H) então cobre todas a arestas da instância (o grafo G) do problema VC.

(b) Sabendo que (VC) é NP-completo (foi provado em aula!), prove que (UFL) também é NP-completo.

Dicas: Observe que, se for interpretado que o (UFL) é definido sobre um grafo, este grafo tem dois tipos de vértices.

(c) Considere a seguinte instância do (VC):  $G = (V, E)$  onde  $V = \{1, 2, 3, 4, 5\}$ , e  $E = \{(1, 2), (1, 4), (2, 3), (3, 4), (1, 5), (2, 5), (3, 5), (4, 5)\}$  e seja  $K = 3$ . Utilize as transformações que você propôs nos itens anteriores para obter instâncias equivalentes do (DS) e do (UFL).

6. Considere o problema (VC) *Vertex Cover* definido na questão anterior. Seja o algoritmo abaixo para resolvê-lo.

- **Algoritmo RVC**

P0 Faça RESPOSTA igual a NÃO.

P1 Para cada subconjunto  $S$  de  $K$  vértices dos vértices em  $V$  faça:

P1.1 Teste se  $S$  é um *Vertex Cover*.

No caso afirmativo faça RESPOSTA igual a SIM, retorne RESPOSTA.

No caso negativo, vá para o próximo subconjunto.

P3 Retorne RESPOSTA.

(a) Demonstre que a complexidade do algoritmo RVC acima é  $O(n^K)$  onde  $n = |V|$ .

(b) Esta complexidade é polinomial ? (Se é, por que não implica que  $P = NP$  ?)