DAS-9003: Introdução a Algoritmos
Prova #2

19 de maio de 2009

Regras:

1. As respostas devem ser escritas à mão.
2. A prova e as respostas devem ser entregues até às 18 h do dia 22/maio (sexta-feira).
3. Os conceitos dependem da corretude das respostas e da eficiência/corretude dos algoritmos projetados.

Questões:

1. Prof. Kunz projetou um algoritmo de ordenação baseado em comparações que executa em tempo $O(n \lg \sqrt{n})$. Você acredita que tal algoritmo existe? Justifique a resposta.
2. Suponha que você tem acesso a uma estrutura de dados D que implementa as operações *search*, *insert*, *delete*, *minimum*, *maximum*, *successor*, e *predecessor* em tempo $O(\lg n)$.

   É possível modificar as operações *insert* e *delete*, de forma que elas continuem sendo executadas em tempo $O(\lg n)$, mas agora *minimum* e *maximum* são executadas em tempo $O(1)$? Se sim, mostre como pode ser realizado. Se não, justifique a impossibilidade.
3. Seja $G = (V, E)$ um grafo direcionado com função peso $w : E \rightarrow \{0, ..., W\}$ onde W é uma constante positiva. Projete um algoritmo que executa em tempo $O(W|V| + |E|)$.
4. Para a questão anterior, projete um algoritmo que executa em tempo $O((|V| + |E|) \lg(|W|))$. Podemos afirmar que este algoritmo é mais rápido que o anterior? Justifique.

5. Descreva um algoritmo que, dado um grafo não-direcionado $G = (V, E)$ e uma função $w : E \rightarrow \mathbb{R}$, encontra uma árvore de espalhamento $T^\star$ tal que o peso da aresta mais pesada de $T^\star$ seja mínimo dentre todas as árvores de espalhamento. Seja $\mathcal{T}$ o conjunto das árvores de espalhamento de G; então o problema pode ser especificado como:

$$P: \quad \max\{w(e) : e \in T^\star\} \quad = \quad \min_{T \in \mathcal{T}} \quad \max\{w(e) : e \in T\}$$

   Qual é o tempo de execução do algoritmo proposto?

6. O Prof. Midas planeja viajar de carro da cidade de Florianópolis/SC até Belém/PA. Ele já estabeleceu uma rota que consiste de trajetos de estradas. Com o tanque de combustível cheio, o carro pode percorrer até n quilômetros. Tendo um mapa com as distâncias entre os postos de gasolina ao longo da rota, o professor deseja determinar os locais onde deve reabastecer tal que nunca lhe falte combustível e o número total de paradas seja o menor possível. Assuma que o Prof. Midas inicia a viagem com o tanque cheio.

   Tarefa:

   (a) projete um algoritmo eficiente para resolver o problema (este deve ter como saída os pontos onde as paradas devem ser realizadas);

   (b) prove que o algoritmos está correto.

7. Mostre que o problema do cômputo da árvore de caminhos mínimos a partir de uma fonte s em um grafo direcionado $G = (V, E)$, com função peso $w : E \rightarrow \mathbb{R}_+$, é $\mathcal{NP}$. Ou seja, desenvolva um algoritmo de verificação rápida: dado uma árvore, o algoritmo deve ser capaz de verificar se esta é uma árvore de caminhos mínimos ou não. Mostre a complexidade do algoritmo. (O algoritmo proposto deve ser o mais eficiente possível.)

8. Seja $X = \{x_1, \bar{x}_1, \ldots, x_m, \bar{x}_m\}$ um conjunto de literais que podem assumir valores verdadeiro/falso. Seja C uma cláusula com n literais, ou seja, C é uma disjunção das variáveis de um subconjunto $S \subseteq X$ com cardinalidade $|S| = n$. Desenvolva um algoritmo para gerar um conjunto de cláusulas $C_1, C_2, \ldots, C_k$, todas com três literais, com a introdução de variáveis y_i $(\bar{y}_i)$ tal que $C_1 \wedge C_2 \wedge \cdots \wedge C_k$ seja equivalente a C.

   Qual a complexidade do algoritmo? Quantas cláusulas são geradas em função de n? Quantas variáveis adicionais são geradas em função de n? (Seu algoritmo deve ser o mais eficiente possível.)

   Tal algoritmo reduz SAT a SAT-3.

9. O problema da árvore de espalhamento de grau mínimo é definido como segue. Dado um grafo não-direcionado G e um inteiro k, G contém uma árvore de espalhamento T tal que o grau de cada vértice u de T é no máximo k? Mostre que este problema é $\mathcal{NP}$-Difícil.

10. Seja S um conjunto com n atividades onde cada atividade j é caracterizada por um instante s_j de início e um instante f_j de término: $s_j, f_j \in \mathbb{N}, j = 1, \ldots, n$, com $s_j < f_j$. Cada atividade j, se alocada, fará uso exclusivo de um recurso durante o intervalo de tempo $[s_j, f_j)$. Duas atividades i e j são ditas compatíveis se $[s_i, f_i) \cap [s_j, f_j) = \emptyset$.

    Tarefas:

    (a) projete um algoritmo eficiente para encontrar um subconjunto $S' \subseteq S$ de atividades compatíveis que maximize a ocupação do recurso compartilhado (tempo total que o recurso é utilizado);

    (b) qual é a complexidade do algoritmo proposto?

    (c) ilustre o funcionamento do algoritmo para a instância dada abaixo.

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14