Universidade Federal de Minas Gerais
Programa de Pós-graduação em Ciência da Computação
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Comissão Permanente de Qualificação 1º Estágio – Curso de Doutorado

Exame de Qualificação 1º Estágio

Área: Teoria

Em 12/03/2008, 9:00 horas.

Prova individual sem consulta com duração de 3 horas

Observações:

1. A prova deve ser resolvida nas folhas distribuídas juntamente com este caderno de questões e rubricadas por um aplicador.
2. A interpretação das questões faz parte da prova. Não será permitida nenhuma pergunta durante a realização desta prova.
3. Caso você ache que falta algum detalhe nos enunciados ou nos esclarecimentos, você deverá fazer as suposições que achar necessárias e escrever essas suposições juntamente com as respostas.
4. Não se esqueça de colocar seu nome em todas as folhas de resposta.

Desejamos que faça uma boa prova!

A COPEQ

Teorema. Sejam as constantes $a \geq 1$ e $b > 1$ e $f(n)$ uma função definida nos inteiros não-negativos pela recorrência:

$$T(n) = aT(n/b) + f(n),$$

onde a fração n/b pode significar $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. A equação de recorrência $T(n)$ pode ser limitada assintoticamente da seguinte forma:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$ e $f(n)$ é polinomialmente menor, então $\boxed{T(n) = \Theta(n^{\log_b a})}$.

2. Se $f(n) = \Theta(n^{\log_b a})$, então $\boxed{T(n) = \Theta(n^{\log_b a} \log n)}$.

3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$ e é polinomialmente maior e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para n suficientemente grande, então $\boxed{T(n) = \Theta(f(n))}$.

Atenção: Escolha 4 dentre as questões 1 a 5, e outras 4 dentre as restantes.

1. Quantas árvores geradoras distintas existem para o grafo K_n? Prove sua resposta.

2. Seja o alfabeto $\Sigma = \{[0,0],[0,1],[1,0],[1,1]\}$. Mostre que é ou não regular o conjunto das palavras $[a_1,b_1][a_2,b_2]\ldots[a_n,b_n]$ em que:

   (a) a seqüência $a_1a_2\ldots a_n$ é o complemento de $b_1b_2\ldots b_n$; exemplo: $[0,1][0,1][1,0]$ pertence ao conjunto, pois 001 é o complemento de 110;

   (b) o número de $\texttt{0}$s de $a_1a_2\ldots a_n$ é igual ao número de $\texttt{0}$s de $b_1b_2\ldots b_n$;

   (c) $a_1a_2\ldots a_n = b_2b_3\ldots b_nb_1$;

   (d) $a_1a_2\ldots a_n = b_nb_{n-1}\ldots b_1$.

3. Sejam $A = \{\texttt{a}\}^*$, $L = \{\texttt{a}^m\texttt{b}^n \mid m \neq n\}$ e $X = (LA) \cup (AL^R)$ (L^R é o reverso de L). Para cada linguagem a seguir, mostre que ela é ou que não é livre do contexto:

   (a) A;

   (b) L;

   (c) X;

   (d) $\overline{X}$.

4. Considere uma máquina de Turing padrão (determinística e com uma única fita, que é limitada à esquerda, mas ilimitada à direita). Para cada caso a seguir, comente o que acontece com seu poder computacional se a máquina for alterada para se tornar *não determinística* e receber também a modificação apontada:

   (a) nenhuma;

   (b) sua fita passa a ser ilimitada também à esquerda;

   (c) sua fita passa a ter exatamente $n+2$ células, sendo n o número de símbolos da palavra de entrada;

   (d) sua fita passa a ter exatamente 2^k células, sendo k o número de estados da máquina.

5. Sejam P_1, P_2 e P_3 problemas de decisão tais que:

   - P_1 é redutível a P_2;
   - P_2 é redutível a P_3;
   - P_3 é redutível a P_1.

   O que se pode dizer quanto à decidibilidade de P_2 e P_3 caso:

   (a) P_1 seja decidível?

   (b) P_1 seja indecidível?

   Justifique com clareza suas respostas: para $i = 1$ e $i = 2$, se P_i for decidível, como poderia ser construído um algoritmo para o mesmo? Se P_i for indecidível, mostre como a existência de um algoritmo para o mesmo levaria a um absurdo.

6. O problema do ciclo mais curto consiste em encontrar o ciclo de menor peso de um grafo ponderado. O ciclo não necessariamente inclui todos os vértices, o que torna esse problema diferente do problema do caixeiro viajante. Determine se o problema do ciclo mais curto é NP-Árduo e justifique sua resposta.

7. Considere um heap binário, com o maior elemento na raiz, representado por um vetor $A[1 \ldots n]$. Suponha que você tenha que mudar o valor do i-ésimo elemento. Apresente um algoritmo eficiente que, após executar $A[i] = x$, atualize o heap adequadamente de forma a restaurar a propriedade do heap.

8. Uma expressão aritmética pode ser representada por uma árvore binária. Por exemplo, a árvore da figura 1 representa a expressão $(2 \times 6 + 3) \times (13 - 8 - 14/7)$.

   Pede-se:

   (a) Escreva um procedimento recursivo baseado em um dos procedimentos de caminhamento em árvores para avaliar uma expressão aritmética. Indique qual caminhamento é esse.

   (b) Escreva a equação de recorrência que representa o custo para resolver este problema no pior caso de acordo com o procedimento apresentado na letra (a).

   (c) Indique qual é a complexidade deste problema no pior caso usando notação O.

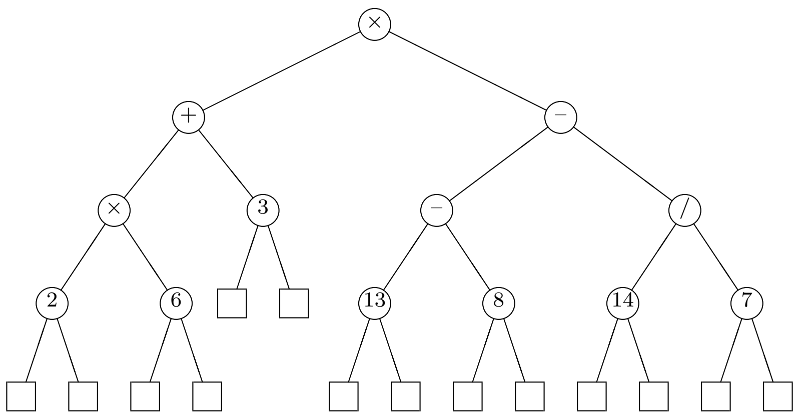


Figura 1: Exemplo de uma árvore binária representando uma expressão aritmética.

9. Use o Teorema Mestre, se for possível, para apresentar o limite assintótico firme para a seguinte recorrência:

$$T(n) = 4T(\frac{n}{2}) + n^2$$

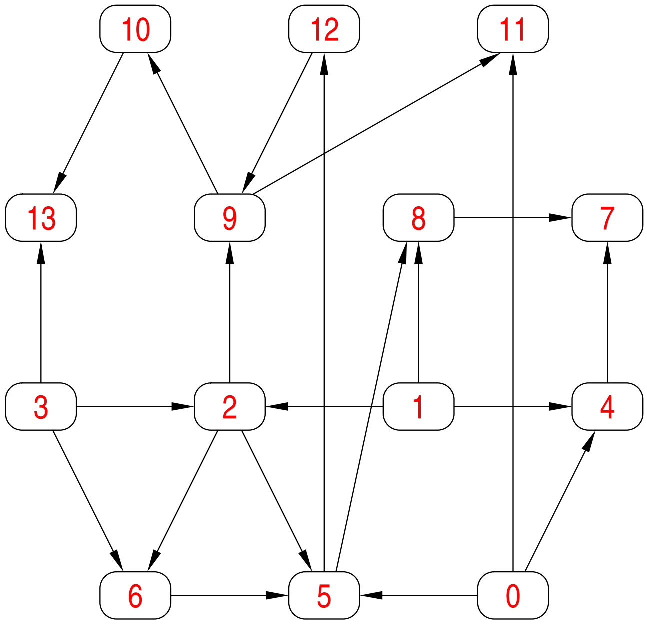
10. (a) O grafo transposto de um grafo direcionado $G = (V, E)$ é definido como o grafo $G^T = (V, E^T)$, onde $E^T = \{(u, v)|(v, u) \in E\}$, isto é, E^T consiste das arestas de G com suas direções invertidas. Discuta a complexidade para gerar o grafo G^T a partir do grafo G supondo que o grafo G está armazenado como:

    (i) Matriz de adjacência.

    (ii) Lista de adjacência.

    (b) Como o número de componentes fortemente conectados muda se uma nova aresta é inserida?

    (c) Mostre a ordem dos vértices produzidos pela ordenação topológica do grafo abaixo.


10
12
11
13
9
8
7
3
2
1
4
6
5
0