Exame Preliminar para o Doutorado
Prova de Análise de Algoritmos

IME–USP, 4 de agosto de 2003
das 14 às 18 horas

Instruções:

(i) O candidato pode resolver todas as questões. A banca considerará questões cujos valores somem até 10 pontos de modo que a soma total das notas obtidas seja máxima.

(ii) Todas as soluções devem conter justificativas das respostas.

(iii) O candidato deve mencionar claramente os teoremas e propriedades usados para justificar suas afirmações.

(iv) Esta prova contém quatro questões de 1 ponto, duas de 2 pontos e duas de 3 pontos.

(v) Comece a responder cada questão em uma folha nova. Não escreva no verso das folhas. Numere as folhas.

Questão 1 [1 ponto] Para um inteiro positivo n, determine o valor do inteiro j, $0 < j < n$, que maximiza $(n-j)^2 + j^2$.

Questão 2 [1 ponto] Prove que $\sum_{i=1}^{n} \frac{i}{2^i} \leq 2$.

Questão 3 [1 ponto] Considere o seguinte problema:

- *dado um vetor $V[1\,..\,n]$ ordenado crescentemente e um número x, decidir se existem índices i e j tais que $V[i] + V[j] = x$.*

Esboce um algoritmo que resolva o problema em tempo $O(n)$.

Questão 4 [1 ponto] Suponha que o número de operações para somar dois números inteiros seja igual à soma dos comprimentos dos números representados na base 2. Determine assintoticamente, em função de n, limitantes inferior e superior para o número de operações para se efetuar as linhas do código abaixo, contabilizando apenas as operações da terceira linha.

```
soma = 0;
para i de 1 até n faça
  soma = soma + i;
```

Questão 5 [2 pontos] Suponha que desejamos não apenas incrementar um contador mas também algumas vezes reinicializá-lo com zero. Mostre como implementar um contador com um vetor binário de maneira que qualquer seqüência de n operações `incrementa1` e `zera_contador` consuma tempo $O(n)$, desde que o contador esteja inicialmente com zero. (**Dica:** Mantenha um apontador para o 1 mais significativo do contador.)

Questão 6 [2 pontos]

1. Explique informalmente o que significa um problema ser NP-completo. Do ponto de vista prático, qual é a relevância de se determinar que um certo problema é NP-completo?

2. Para cada uma das afirmações abaixo, diga se ela é *verdadeira*, *falsa*, *verdadeira se* $P \neq NP$ ou *falsa se* $P \neq NP$.

    1. Não há problemas em P que são NP-completos.
    2. Existe apenas algoritmo exponencial para o problema da parada.
    3. Existem problemas em P que estão em NP.
    4. Se A pode ser polinomialmente reduzido a B e B está em P então A está em P.
    5. Se A pode ser polinomialmente reduzido a B e B é NP-completo então A é NP-completo.

    Dê uma justificativa **curta** para cada resposta.

Questão 7 [3 pontos] Sejam $X[1..n]$ e $Y[1..n]$ dois vetores ordenados crescentemente. Dê um algoritmo $O(\log n)$ para encontrar a mediana de todos os $2n$ elementos nos vetores X e Y. (A *mediana* de um vetor ordenado $Z[1..m]$ é o elemento $Z[j]$, onde $j = \lceil m/2 \rceil$.)

Questão 8 [3 pontos] Seja $\{1, \ldots, n\}$ um conjunto de *tarefas*. Cada tarefa consome um dia de trabalho; durante um dia de trabalho somente uma das tarefas pode ser executada. Os dias de trabalho são numerados de 1 a n. A cada tarefa t está associado um *prazo* p_t: a tarefa deveria ser executada em algum dia do intervalo $1 \mathinner{..} p_t$. A cada tarefa t está associada uma *multa* não-negativa m_t. Se uma dada tarefa t é executada depois do prazo p_t, sou obrigado a pagar a multa m_t (mas a multa não depende do número de dias de atraso). Problema: programar as tarefas (ou seja, estabelecer uma bijeção entre as tarefas e os dias de trabalho) de modo a minimizar a multa total. Escreva um algoritmo para resolver o problema. Prove que seu algoritmo está correto. Analise o consumo de tempo.