# Exame Preliminar para o Doutorado
# Prova de Análise de Algoritmos

IME–USP, 23 de agosto de 2000
das 14 às 18 horas

**Instruções**: *(1)* 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.
*(2)* O candidato deve mencionar claramente os teoremas e propriedades usados para justificar suas afirmações.
*(3)* Esta prova contém uma questões de 1 ponto, cinco de 2 pontos e uma de 3 pontos.
*(4)* Comece a responder cada questão em uma folha nova. Não escreva no verso das folhas.

**Questão 1** [1 ponto] Suponha que $T$ é uma função que leva números naturais em números naturais. Sabe-se que

$$T(n) \leq T(n-1) + O(n)\,.$$

Discuta o significado dessa afirmação. A que ordem $O$ a função $T$ pertence? Justifique.

**Questão 2** [2 pontos] Duas estruturas que podem ser usadas para implementar filas de prioridades são o heap e a árvore balanceada binária de busca. Compare as duas implementações em termos da eficiência das operações básicas, e descreva situações que favoreceriam usar uma ou outra.

**Questão 3** [2 pontos] Uma seqüência de $n$ operações é executada sobre uma certa estrutura de dados. Suponha que a $i$-ésima operação custa

$i$ se $i$ é uma potência de 2 e
1 em caso contrário.

1. Determine o custo amortizado de uma operação. Justifique.

2. Dê uma função potencial apropriada para a estrutura de dados e deduza daí o custo amortizado de uma operação. Justifique.

**Questão 4** [2 pontos]   Suponha que um vetor $A[1\,..\,n]$ está organizado como um heap. A *altura* de um nó $i$ no vetor é definida da maneira usual: trata-se do comprimento do mais longo caminho da forma $i, f(i), f(f(i)), \ldots$, onde $f(i)$ vale $2i$ ou $2i{+}1$. Assim, por exemplo, a altura do nó $n$ é 0. É verdade que a altura de um nó $i$ é

$$\left\lfloor \lg \frac{n}{i} \right\rfloor \text{ ?}$$

Justifique.

**Questão 5** [2 pontos]   Em situações onde é necessário desperdiçar tempo com classe (por exemplo, se você cobra por hora), é necessário usar algoritmos que sejam provadamente ineficientes inclusive no melhor caso. Por exemplo, um bubblesort não é eficientemente ineficiente, pois se os dados estiverem já em ordem, o algoritmo termina em $O(n)$ passos.

SLOWSORT $(i, j)$   $\triangleright$ ordena o trecho $A[i], A[i{+}1], \ldots, A[j]$ do vetor global $A$
    **se** $i < j$
        **então** $m \leftarrow \lfloor (i + j)/2 \rfloor$
                SLOWSORT $(i, m)$
                SLOWSORT $(m{+}1, j)$
                **se** $A[m] \geq A[j]$ **então** $A[m] \leftrightarrow A[j]$   $\triangleright$ troca $A[m]$ com $A[j]$
                SLOWSORT $(i, j{-}1)$

Mostre que o algoritmo está correto, e seu desempenho é $\Omega(n^9)$, onde $n$ é o número de elementos de $A$.

**Questão 6** [2 pontos]   Considere o seguinte algoritmo, que recebe um vetor $A[1\,..\,n]$ e devolve o índice de um elemento mínimo do vetor:

MÍNIMO $(A, n)$
1   **se** $n = 1$
2       **então** $k \leftarrow n$
3       **senão** $k \leftarrow$ MÍNIMO $(A, n{-}1)$
4             **se** $A[k] > A[n]$
5                 **então** $k \leftarrow n$
6   **devolva** $k$

Suponha que os elementos de $A[1\,..\,n]$ são distintos dois a dois. Suponha também que, para cada $i$, a probabilidade de que $A[i]$ é o elemento mínimo do vetor é $1/n$.

Qual o número médio, digamos $T(n)$, de execuções do comando "$k \leftarrow n$" (linhas 2 e 5). [*Sugestão*: Escreva uma recorrência para $T(n)$.]

**Questão 7** [3 pontos]   Suponha que um vetor $A[1\,..\,n]$ está organizado como um heap "crescente" (portanto $A[1]$ é um elemento mínimo). Considere o seguinte problema:

> dado um índice $i$, remover $A[i]$ e transformar o vetor restante
> em um heap $A[1\,..\,n{-}1]$.

Descreva um algoritmo que resolva o problema em tempo $O(\lg n)$. Justifique sua solução.