Exame Preliminar para o Doutorado
Prova de Análise de Algoritmos

IME–USP, 5 de março de 2001
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) Todas as soluções das questões de um ponto devem conter uma justificativa sucinta da resposta. Uma frase *bem escolhida* em cada questão está de bom tamanho.
(4) Esta prova contém oito questões de 1 ponto e três de 2 pontos.
(5) Comece a responder cada questão em uma folha nova. Não escreva no verso das folhas.

Questão 1 [1 ponto]
Seja f uma solução da recorrência

$$f(2n+1) = f(2n) = f(n) + \log n, \quad \text{para } n \geq 1,$$

com $f(1) = 0$. Qual dos seguintes dá o melhor limite superior para $f(n)$?
(a) $O(\log n)$ (b) $O(n \log n)$ (c) $O(\log n) + O(1)$ (d) $O((\log n)^2)$ (e) $O(n)$
Justifique **concisamente** a sua resposta.

Questão 2 [1 ponto]

Considere um tipo de dados cujos elementos são inteiros e cujas operações são INSERÇÃO, REMOÇÃO e ACHA-MAIS-PRÓXIMO, com ACHA-MAIS-PRÓXIMO(y) definido como algum elemento x no conjunto corrente tal que $|x - y| \leq |z - y|$ para todo z no conjunto corrente. Seja

$$T := \max(T_{\text{INSERÇÃO}}, T_{\text{REMOÇÃO}}, T_{\text{ACHA-MAIS-PRÓXIMO}}),$$

onde T_{OP} denota a complexidade de tempo de pior caso para a operação OP. Qual das seguintes estruturas de dados seria melhor para minimizar T?

(a) Um vetor ordenado;
(b) Um vetor não-ordenado;
(c) Um *heap*;
(d) Uma árvore balanceada;
(e) Uma tabela de espalhamento (*hashing*) com conflitos resolvidos por lista ligada.

Justifique **concisamente** a sua resposta.

Questão 3 [1 ponto]

Qual é o maior número possível de nós internos em uma árvore rubro-negra de altura k?

Justifique **concisamente** a sua resposta.

Questão 4 [1 ponto]

Suponha que um certo problema X depende de um parâmetro inteiro positivo n. Meu algoritmo para o problema consome tempo $O(n^2)$. Meu amigo diz ter um algoritmo melhor, que consome tempo $\Omega(n)$. Devo ficar impressionado?

Justifique **concisamente** a sua resposta.

Questão 5 [1 ponto]

Meu manual diz que um certo problema "está em NP"; eu concluo que não existe algoritmo polinomial para o problema. Estou certo ou errado?

Justifique **concisamente** a sua resposta.

Questão 6 [1 ponto]

Suponha dado um grafo conexo cada uma de cujas arestas tem um peso numérico. Discuta o seguinte algoritmo, que promete encontrar uma árvore geradora de peso mínimo no grafo:

> Para cada vértice v, escolha, dentre as arestas incidentes a v, uma que tenha peso mínimo.

Questão 7 [1 ponto]

Suponha dadas n chaves entre 1 e n^2. Mostre que é possível construir uma árvore de busca binária com essas chaves em tempo $O(n)$.

Questão 8 [1 ponto]

Seja $A[1..n]$ um vetor de n números reais, onde $A[i] = \log_2(i/(i+1))$, para cada i entre 1 e n. Mostre que o vetor A é um heap do tipo em que $A[1]$ é o elemento máximo.

Justifique **concisamente** a sua resposta.

Questão 9 [2 pontos]

Um *segmento* de um vetor $A[1..n]$ é qualquer vetor da forma $A[p..q]$ com $1 \leq p \leq q \leq n$. O *comprimento* de um tal segmento é $q - p + 1$. O segmento é *crescente* se $A[p] \leq \cdots \leq A[q]$.

Problema: Dado um vetor $A[1..n]$, encontrar o comprimento de um segmento crescente de comprimento máximo.

1. Escreva um algoritmo *recursivo* eficiente que resolva o problema.

2. Escreva a recorrência que define o consumo de tempo de pior caso do algoritmo (suponha que cada comando "simples" do algoritmo consome uma unidade de tempo).

3. Resolva a recorrência do item anterior, determinando assim o consumo de tempo do algoritmo no pior caso.

Questão 10 [2 pontos]

Uma loja de aluguel de esquis tem m pares de esquis para alugar, sendo s_i a altura do i-ésimo par de esquis. Existem também n esquiadores querendo alugar pares de esquis, sendo h_i a altura do i-ésimo esquiador. Sabe-se que, idealmente, cada esquiador deveria usar um esqui cuja altura é tão próxima quanto possível da altura do próprio esquiador. Supondo que $n = m$, escreva um algoritmo eficiente de alocação de esquis aos esquiadores de forma a minimizar a soma dos valores absolutos das diferenças entre as alturas do esquiador e do seu esqui. Analise a eficiência do seu algoritmo e justifique sua corretude.

Sugestão: Mostre que existe uma solução ótima sem cruzamentos de alturas, i.e., se esquiadores i e j têm alturas $h_i < h_j$ seus respectivos esquis terão alturas $s_i < s_j$.

Observação: Note que a escolha dos pares (esqui,esquiador) fica muito mais complicada sem a hipótese de n e m serem iguais.

Questão 11 [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)    ▷ ordena o trecho A[i], A[i+1], ..., A[j] do vetor global A
  se i < j
    então m ← ⌊(i + j)/2⌋
          Slowsort (i, m)
          Slowsort (m+1, j)
          se A[m] ≥ A[j]
            então A[m] ↔ A[j]   ▷ troca A[m] com A[j]
          Slowsort (i, j−1)
```

Mostre que o algoritmo está correto, e seu desempenho é $\Omega(n^7)$, onde n é o número de elementos de A.

BOA SORTE!