EXAME PRELIMINAR PARA O DOUTORADO

IME–USP, 22 de Agosto de 2001
das 14 às 18 horas

Prova de Análise de Algoritmos

Instruções:

(i) O candidato pode resolver todas as questões.

(ii) A banca considerará questões cujos valores somem até 10 pontos de modo que a nota seja máxima.

(iii) Todas as soluções das questões de 1 ponto devem conter uma justificativa sucinta da resposta. Uma frase *bem escolhida* em cada questão está de bom tamanho.

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

Questão 1 [1 ponto]

Quantas vezes a comparação "$A[r] \neq 0$" é executada? Defina esse número por meio de um recorrência.

```
LIMPA (A, p, r)
se p ≥ r
    então devolva r
    senão q ← LIMPA (A, p, r − 1)
          se A[r] ≠ 0
             então q ← q + 1
                   A[q] ← A[r]
          devolva q
```

Dê uma fórmula exata para a função definida pela recorrência. Em que classe Θ está a função definida pela recorrência? Explique.

Questão 2 [1 ponto]

A afirmação abaixo é verdadeira ou falsa? Comente e justifique.

> "Qualquer algoritmo que rearranje um vetor $A[1 \mathinner{..} n]$ de modo que ele se torne um heap consome $\Omega(n \lg n)$ unidades de tempo."

Questão 3 [1 ponto]

Prove que

> *Se NP $\neq$ co-NP, então P $\neq$ NP.*

Questão 4 [1 ponto]

Se $f(n) = \mathrm{O}(n)$ e $g(n) = \mathrm{O}(n^2)$, qual dessas funções cresce mais rápido? Justifique sua resposta.

Questão 5 [2 pontos]

Descreva informalmente (não escreva código) um algoritmo que consome $O(n)$ unidades de tempo para imprimir os $\lfloor\sqrt{n}\rfloor$ maiores elementos de um vetor $A[1\,..\,n]$. Explique porque o algoritmo funciona e a análise de tempo.

Questão 6 [2 pontos]

Dê limitantes inferiores e superiores para a altura de uma B-árvore em que cada nó tem entre 100 e 200 filhos. Explique seus cálculos.

Questão 7 [3 pontos]

Considere a seqüência de vetores

$$A_k[1\,..\,2^k],\ A_{k-1}[1\,..\,2^{k-1}],\ \ldots,\ A_1[1\,..\,2^1],\ A_0[1\,..\,2^0]\,.$$

Suponha que cada um dos vetores é crescente. Queremos reunir, por meio de sucessivas operações de intercalação (= *merge*), o conteúdo dos vetores $A_0, \ldots, A_k$ em um único vetor crescente $B[1\,..\,n]$, onde $n = 2^{k+1} - 1$. Escreva um algoritmo que faça isso em $O(n)$ unidades de tempo. Você não precisa escrever o código da rotina INTERCALA, mas precisa dizer *o que* ela faz exatamente. Justifique.

Questão 8 [3 pontos]

O código abaixo descreve a implementação de uma fila usando duas pilhas, `E` e `D`. As funções `push`, `pop`, `empty`, respectivamente, empilham, desempilham e testam se a pilha é vazia, e são consideradas operações elementares. A fila vazia é representada com ambas as pilhas vazias; o teste para fila vazia não está descrito.

```
void insere (int x) {
  push(D, x)
}

int remove {
  if (empty(E)) {
    if (empty(D)) {if (windows) {show blue screen of death} else {exit gracefully}}
    while (!empty(D))
      push(E, pop(D))
  }
  return pop(E)
}
```

Analise o custo amortizado das operações `insere` e `remove` para uma sequência de inserções e remoções de n elementos, que termine com a fila vazia. Note que a sequência de operações é arbitrária, exceto que não vale tentar remover um elemento da fila vazia.