Exame — MAC5811-1 – Projeto e Análise de Algoritmos

2/3/2004

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*) Algoritmos devem ser escritos em pseudocódigo ou de forma informal usando algoritmos bem conhecidos como subrotinas.

(*iv*) Explique sempre porque cada algoritmo funciona e estime sua complexidade.

(*v*) A duração da prova é ∞ horas (ou até o `cef` cansar de tomar conta, o que acontecer primeiro).

Questão 1 [1 ponto]

Considere o seguinte método que determina o maior elemento de um vetor:

```
int maximo (vetor v, int n)
{
   int max;

   if (n == 1)
     max = n-1;    /* (*)  */
   else{
     max = maximo (v, n-1);
     if (v[max] < v[n-1])
       max = n-1; /* (*) */
   }

   return(max);
}
```

(a) Escreva uma recorrência que descreva o número médio de vezes que as as atribuições `(*)` são executadas, supondo que todos os elementos do vetor têm igual probabilidade de ser o máximo.

(b) Resolva a recorrência do item (a).

Questão 2 [1 ponto]

Para o problema: *dadas listas ordenadas A e B de elementos distintos, decidir se todo elemento de A é também elemento de B*, apresente um algoritmo baseado em comparações que leve tempo linear no total de elementos das duas listas.

Questão 3 [1 ponto]

Para que valores reais da constante α é verdade que n^α cresce mais que $\lg n$? Em outras palavras, para que valores reais de α vale: $\lg n = o(n^\alpha)$?

Questão 4 [1 ponto]

P=NP se e só se algum problema em P é NP-completo.
Essa afirmativa é verdadeira ou falsa? Justifique.

Questão 5 [2 pontos]

Descreva um algoritmo de programação dinâmica que, dados um conjunto A com n inteiros positivos e um inteiro S, determina em tempo $O(nS)$ se existe algum subconjunto de A cuja soma é S.

Questão 6 [2 pontos]

Vamos definir a função $d(x, y, z)$ = *x-y* + *y-z* + *z-x*, onde x é o valor absoluto do inteiro x.
Seja A um vetor de n inteiros, não ordenado. O problema é detrminar três elementos distintos x, y, z que minimizem $d(x, y, z)$.

(a) Dê pseudocódigo para um algoritmo simples que resolva o problema em tempo $O(n^3)$.

(b) Descreva um algoritmo que resolva esse problema de forma mais eficiente, isto é, em tempo $o(n^3)$. Dica: qual o valor de $d(x, y, z)$ quando $x < y < z$?

Questão 7 [3 pontos]

Uma árvore A é *zona* de uma árvore B (esse termo é usado no DNS) se A pode ser obtida de uma sub-árvore de B pela remoção de algumas sub-árvores. Dê um algoritmo eficiente para, dadas árvores de busca binária A e B, decidir se A é zona de B. Analise a complexidade do seu algoritmo em termos dos tamanhos e alturas das duas árvores.

Questão 8 [3 pontos]

O algoritmo de Kruskal tem como entrada um grafo com pesos nas arestas, e devolve uma árvore geradora de peso mínimo. Considere a seguinte implementação desse algoritmo:

1. Inicia-se com uma ‘”árvore” (na verdade, uma floresta) vazia.
2. As arestas são colocadas num heap para seleção de mínimo.
3. Os vértices formam uma partição, cujos blocos são unitários.
4. As arestas são selecionadas em ordem crescente de peso; quando suas duas pontas pertencem a blocos diferentes da partição, os blocos são unidos e a aresta é adicionada à árvore.
5. O algoritmo pára quando todos os vértices ficam no mesmo bloco.

Suponha que é dado para seu algoritmo o grafo completo com vértices $\{1, 2, \ldots, n\}$, em que o peso de uma aresta ij é:

1. $\min(i, j) + n * \max(i, j)$.
2. $\max(i, j) + n * \min(i, j)$.

Em cada caso, estime, em função de n, o tempo gasto pelo algoritmo.