Exame de MAC5811 - Projeto e Análise de Algoritmos

DCC–IME–USP, fevereiro, 2006

Instruções:

(*i*) O candidato pode resolver todas as questões.

(*ii*) Esta prova contém quatro questões de um ponto, duas de dois pontos e duas de três pontos.

(*iii*) A banca considerará questões cujos valores somem até 10 pontos de modo que a soma total das notas obtidas seja máxima. Um aluno, para ser aprovado, precisa obter nessas questões pelo menos 7 pontos.

(*iv*) Você pode utilizar algoritmos bem conhecidos como subrotina de seus algoritmos.

(*v*) Ao escrever seus algoritmos, coloque comentários que ajudem na compreensão do que eles fazem mesmo na presença de eventuais erros de sintaxe.

(*vi*) Dê justificativas sucintas de que cada um de seus algoritmos e suas análises estão corretos.

Duração do exame: 5 horas

1. **(Valor: 1 ponto)** Mostre que $\sum_{i=1}^{n} i^2 = \Theta(n^3)$.

2. **(Valor: 1 ponto)** Considere a seguinte função recursiva que determina o índice de um maior elemento de um vetor $v[1\,..\,n]$:

```
IND_MAXIMO(v, n)
1  se n = 1
2     então imax ← n
3     senão imax ← IND_MAXIMO(v, n − 1)
4           se v[imax] < v[n]
5              então imax ← n
6  devolva imax
```

   Supondo que a entrada é uma permutação aleatória de $\{1, \ldots, n\}$, escolhida de acordo com a distribuição uniforme, qual é o número esperado de vezes que as atribuições das linhas 2 e 5 são executadas? Justifique sua resposta.

3. **(Valor: 1 ponto)** Em uma lanchonete o cozinheiro era um fanático por algoritmos de ordenação que não arrumou emprego e acabou tendo de fazer panquecas. Mesmo longe de seu computador ele não consegue se desligar dos algoritmos. Todos os dias, depois de fritar as panquecas, ele as ordena em ordem decrescente de diâmetro, ou seja, as maiores embaixo e as menores em cima. A única operação de que ele dispõe é a operação de enfiar a espátula em uma posição qualquer da pilha de panquecas e inverter toda a pilha dali para cima. Com isso, as panquecas abaixo da espátula permanecem em suas posições, a panqueca que estava imediatamente em cima da espátula vai parar no topo, a que estava a seguir vai parar embaixo dessa, e assim por diante, até a panqueca que estava no topo.

Suponha que são dados os diâmetros das n panquecas em um vetor $v[1\,..\,n]$ (a panqueca na primeira posição do vetor está no fundo da pilha) e suponha que o único método disponível é $\textsc{flip}(v,k,n)$ que inverte a ordem das panquecas como descrito acima a partir da posição k (assim, $\textsc{flip}(v,1,n)$ inverteria toda a pilha). Escreva um algoritmo para ordenar as panquecas que faça $O(n)$ chamadas ao método FLIP.

4. **(Valor: 1 ponto)** Dê um algoritmo eficiente para: dado um grafo conexo G com n vértices, m arestas e pesos não-negativos nas arestas, encontrar um conjunto de arestas cuja remoção não desconecta G, e que tenha peso máximo. Como sempre, o peso de um conjunto de arestas é a soma de seus pesos. Note que estamos falando de remoção de arestas, sem remoção de vértices.

5. **(Valor: 2 pontos)** Resolva as seguintes recorrências. (Você pode assumir que n é potência de 2.)

   (a) $T(n) = 4T(n/2) + n^2$

   (b) $T(n) = 4T(n/2) + n^2 \lg n$

6. **(Valor: 2 pontos)** Considere esses dois problemas, em que é dado um grafo G (com n vértices) e dois vértices s e t:

   (P1) Determinar se existe um caminho de s a t que passa por no máximo $n/2$ vértices.

   (P2) Determinar se existe um caminho de s a t que passa por no mínimo $n/2$ vértices.

   Considere as seguintes afirmativas:

   (a) Não existe algoritmo polinomial para nenhum desses problemas.

   (b) Existe algoritmo polinomial para pelo menos um desses problemas.

   (c) Existe algoritmo polinomial para exatamente um desses problemas.

   (d) Existe algoritmo polinomial para ambos problemas.

   Quais dessas afirmativas são verdadeiras ou falsas com certeza? Se o problema $P = NP$ for resolvido, no que isso afeta a resposta à pergunta anterior?

7. **(Valor: 3 pontos)** Você deve cortar uma tora de madeira em vários pedaços. A empresa mais em conta para fazer isso é a *Cortadora de Pedaços Inteiros (CPI)*, que cobra de acordo com o comprimento da tora a ser cortada. A máquina de corte deles permite que apenas um corte seja feito por vez.

   Se queremos fazer vários cortes, é fácil ver que ordens diferentes destes cortes levam a preços diferentes. Por exemplo, considere uma tora com 10 metros de comprimento, que tem que ser cortada a 2, 4 e 7 metros de uma de suas extremidades. Há várias possibilidades. Podemos primeiramente fazer o corte dos 2 metros, depois dos 4 e depois dos 7. Tal ordem custa $10 + 8 + 6 = 24$, porque a primeira tora tinha comprimento 10, o que restou tinha 8 metros de comprimento e o último pedaço tinha comprimento 6. Se cortássemos na ordem 4, depois 2, depois 7, pagaríamos $10 + 4 + 6 = 20$, que é mais barato.

   Escreva um algoritmo que, dado o comprimento l da tora, um inteiro não-negativo k e um vetor $p[1\,..\,k]$ de inteiros tal que $0 < p[1] < p[2] < \cdots < p[k] < l$, com os pontos onde a tora deve ser cortada, encontre o custo mínimo para executar esses cortes na CPI. Seu algoritmo deve consumir tempo $O(k^3)$.

8. **(Valor: 3 pontos)** Uma matriz $m \times n$ *semi-ordenada* é uma matrix $m \times n$ tal que as entradas de cada linha estão em ordem não-decrescente da esquerda para a direita, e as entradas de cada coluna estão em ordem não-decrescente de cima para baixo. Algumas entradas de uma matriz semi-ordenada podem valer ∞, e são tratadas como posições vagas da matriz. Assim uma matriz semi-ordenada pode ser usada para armazenar um conjunto de $r \leq mn$ números. Dizemos então que uma matriz semi-ordenada $Y_{m \times n}$ está *vazia* se $Y[1,1] = \infty$ e que está *cheia* se $Y[m,n] < \infty$.

   (a) Desenhe uma matriz semi-ordenada 3×3 com os números do conjunto $\{9, 16, 3, 2, 4, 8, 5, 14, 12\}$.

   (b) Escreva um algoritmo EXTRAI_MIN que recebe como parâmetro uma matriz semi-ordenada $A_{m \times n}$ e extrai da matriz A um elemento de valor mínimo armazenado na matriz e o devolve. (Devolve ∞ se a matriz está vazia.) Após a extração desse elemento, a matriz A deve continuar sendo uma matriz semi-ordenada. Seu algoritmo deve consumir tempo $O(m + n)$. (*Dica:* Pense no algoritmo de extração de mínimo de um heap.)

   (c) Escreva um algoritmo INSERE que recebe como parâmetros um número x e uma matriz semi-ordenada $A_{m \times n}$, que não está cheia, e insere x em A, mantendo-a uma matriz semi-ordenada. Seu algoritmo deve consumir tempo $O(m + n)$.

   (d) Sem usar um outro algoritmo de ordenação, explique como usar uma matriz semi-ordenada $n \times n$ para ordenar n^2 números em tempo $O(n^3)$.

   (e) Escreva um algoritmo BUSCA que recebe como parâmetros um número x e uma matriz semi-ordenada $A_{m \times n}$ e determina se x está em A ou não. Seu algoritmo deve consumir tempo $O(m + n)$.

BOA PROVA!