PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2006.1
Horário: 2as-feiras e 4as-feiras de 11 às 13 horas - Sala 774L

PROJETO E ANÁLISE DE ALGORITMOS (INF 2926)

Lista Obrigatória 1

1. Considere uma *heap* com n inteiros e um valor x. A *heap* possui o seu maior elemento no topo e está organizada em um vetor (de n posições). Proponha um algoritmo para decidir se o valor x é maior ou igual ao k-ésimo maior elemento na *heap* ou não. O algoritmo deve executar em $O(k)$.

   Dica: Observe que você não precisa necessariamente encontrar o k-ésimo maior elemento da *heap*.

2. Considere o seguinte problema:

   [ELE] Dados um conjunto de n inteiros distintos $X = \{x_1, \ldots, x_n\}$ e pesos positivos $w(x_j)$, $j = 1, \ldots, n$, associados aos elementos de X, e um inteiro positivo V tal que $0 \leq V \leq W = \sum_{j=1}^{n} w(x_j)$; encontrar o elemento x_k tal que $\sum_{x_j < x_k} w(x_j) < V$ e $w(x_k) + \sum_{x_j < x_k} w(x_j) \geq V$.

   (a) Projete um algoritmo $O(n\,log\,n)$ para resolver esse problema.
   (b) Utilize divisão e conquista para projetar um algoritmo $O(n)$ para resolver esse problema.
   (c) Apresente detalhadamente a análise da complexidade do item anterior.

3. Sejam u e v dois números de n bits (considere que n é potência de 2). A multiplição tradicional de u por v utiliza $O(n^2)$ operações. Um algoritmo baseado em divisão e conquista divide os números em duas partes iguais, calculando o produto como: $uv = (a2^{n/2} + b)(c2^{n/2} + d) = ac2^n + (ad + bc)2^{n/2} + bd$. As multiplicações ac, ad, bc e bd são feitas usando este mesmo algoritmo recursivamente.

   - Escreva este algoritmo
   - Determine a complexidade
   - Qual é a complexidade se $ad + bc$ é calculado como $(a + b)(c + d) - ac - bd$?

4. Verdadeiro ou Falso. Justifique.

   (a) $(log\,n)^{100} = O(n^{\epsilon})$, $\forall \epsilon > 0$.
   (b) $2^{n+1} = O(2^n)$.
   (c) Se $g(n, m) = m log_d n$ onde $d = \lceil m/n + 2 \rceil$ ($\lceil x \rceil$ é o menor inteiro maior que x), $m = O(n^2)$, então $g(n, m) = O(m^{(1+\epsilon)})$ $\forall \epsilon > 0$.

5. Determine uma forma fechada para cada uma das seguintes recorrência:

   (a) $T(1) = 1$;
   $T(2) = 6$;
   $T(n) = T(n-2) + 3n + 4$, $\forall n \geq 3$.

(b) $T(1) = 1$;
$T(2) = 6$;
$T(n) = 2.T(n-2) + 3, \forall n \geq 3.$

(c) $T(1) = 1$;
$$T(n) = \sum_{i=1}^{n-1}(T(i) + T(n-i)) + 1, \forall n \geq 2.$$

6. Seja $S = \{x_1, x_2, \ldots, x_n\}$ uma seqüência de números reais (não necessariamente positivos). Projete dois algoritmos de complexidade $O(n)$, um iterativo e um que utilize divisão e conquista (ambos podem ser recursivos!), que determine uma subseqüência $S' = \{x_i, x_{i+1}, \ldots, x_j\}$ de elementos consecutivos da seqüência S tal que o produto dos números que compõem S' é máximo dentre todas as subseqüências consecutivas possíveis.

7. Dados um multiconjunto C contendo n números reais (não necessariamente distintos) e um número real x:

   (a) Projete um algoritmo de complexidade $O(n \log n)$ que determine se existem dois elementos em C cuja soma seja igual a x.

   (b) Considere agora que os elementos do conjunto C estão ordenados crescentemente. Projete um algoritmo de complexidade $O(n)$ para resolver o mesmo problema.

8. O departamento de transporte de uma cidade deseja averiguar se seus motoristas respeitam os conceitos pregados pela técnica de direção defensiva. De acordo com o departamento, todos os veículos deveriam manter entre si uma distância maior ou igual d, definida como *distância de segurança*. Com o objetivo de realizar uma pesquisa, várias câmeras foram instaladas pela cidade, captado as posições de inúmeros veículos. A posição de um veículo é definida por um par ordenado $(x, y) \in R^2$. Seu objetivo é projetar um algoritmo de complexidade $O(n \log n)$ que obtenha uma cena contendo as posições de n veículos captadas por uma câmera e a distância de segurança d atualmente estabelecida pelo departamento, informando ao final da execução se existem veículos que não respeitam a distância de segurança. Seu algoritmo deve responder apenas *sim* ou *não*. A distância entre dois veículos com posições (x_1, y_1) e (x_2, y_2) é dada por $\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$.

9. O problema da torre de Hanói generalizado consiste em mover n discos de diâmetros distintos, posicionados em um haste denominada *origem*, para uma haste denominada *destino* utilizando $m \geq 1$ hastes denominadas *de trabalho*. Inicialmente, todos os discos encontram-se empilhados na haste de origem em ordem decrescente de tamanho, de baixo para cima. As demais hastes de trabalho e destino encontram-se vazias. Durante o processo de transferência é permitida a movimentação de apenas um disco por vez. Considere ainda que nenhum disco pode ser posicionado acima de um disco com diâmetro menor que o seu. Projete um algoritmo que resolva o problema da torre de Hanói generalizado utilizando o menor número de movimentos possível. Seu algoritmo deve relatar a ordem e a quantidade de movimentos a serem realizados.