DINTER UFF/IFTM - Análise e Síntese de Algoritmos - Lista de Exercícios - 2013

1. Uma pessoa sobe uma escada composta de n degraus, com passos que podem alcançar entre 1 e $k \leq n$ degraus. Escrever equações de recorrência que permitem determinar o número de modos distintos de a pessoa subir a escada.

2. Escrever equações de recorrência que fornecem o número de maneiras distintas de efetuar operações binárias em uma sequência de n operandos (isto é, colocar parênteses).

3. Escrever equações de recorrência que fornecem o número de maneiras distintas de se triangularizar um polígono convexo de n lados, $n > 3$. Assuma que os vértices têm rótulos $1, 2, \ldots, n$.

4. Resolva a recorrência abaixo:

   $T(1) = 1$,
   $T(n) = 3\,T(n-1) + 2$, para todo $n > 1$.

   (a) Inicialmente, use o método da força bruta para descobrir uma fórmula fechada para $T(n)$.

   (b) A seguir, prove por indução que a fórmula está correta.

5. Resolver o problema da Torre de Hanói quando se dispõe de 4 varetas, em vez de 3. Determinar o número de movimentos de disco.

6. Um **torneio** é um grafo direcionado onde, para qualquer par de vértices a, b, exatamente uma das arestas direcionadas (a, b) ou (b, a) existe. Um **caminho hamiltoniano** num grafo direcionado é um caminho direcionado que contém cada vértice do grafo exatamente uma vez. Prove que todo torneio tem um caminho hamiltoniano. A sua prova deve ser um algoritmo recursivo que constrói um caminho hamiltoniano para um torneio.

7. Considere o seguinte algoritmo:

   Entrada: vetor $L[1 \cdots n]$ contendo n elementos
   para $i = 1 \ldots n-1$ faça
   $j \leftarrow i + 1$
   enquanto $j \leq n$ faça
   se $L[i] > L[j]$ então troque os elementos $L[i]$ e $L[j]$
   $j \leftarrow j + 1$

   (a) Qual é a complexidade de *pior caso* do algoritmo acima? (Baseada em *número de trocas entre elementos.*) Descreva uma entrada com $n = 5$ elementos que leva o algoritmo ao *pior caso*, mostrando as trocas de elementos efetuadas.

   (b) Idem, mas agora para *melhor caso.*

8. Considere o seguinte algoritmo de ordenação:

   função ordena(L)
   se $|L| \leq 1$ então retornar L
   senão
   $x \leftarrow$ primeiro elemento de L
   $L_1 \leftarrow$ vetor formado pelos elementos de L menores do que x
   $L_2 \leftarrow$ vetor formado pelos elementos de L iguais a x
   $L_3 \leftarrow$ vetor formado pelos elementos de L maiores do que x
   retornar a concatenação dos vetores: ordena(L_1), L_2, ordena(L_3)

(a) Mostre a árvore de recursão deste algoritmo para a entrada $L = [5, 2, 1, 7, 9, 3, 1, 6]$. Cada nó desta árvore é uma chamada recursiva. Ao definir os vetores L_1, L_2, L_3, respeitar a mesma ordem relativa que os elementos têm em L.

(b) Analise a complexidade de pior caso deste algoritmo para um vetor L com n elementos. (A operação básica a ser considerada no cálculo é a comparação entre dois elementos.) Mostre uma entrada com $n = 8$ que leva o algoritmo ao pior caso.

9. Considere o seguinte algoritmo:

função mistério(y, z)
entrada: números naturais y e z
$x \leftarrow 0$;
enquanto $z > 0$ **faça**
    se z mod $2 = 1$ **então** $x \leftarrow x + y$ **fim-se**; (z mod 2 é o resto da divisão de z por 2)
    $y \leftarrow 2y$; $z \leftarrow \lfloor z/2 \rfloor$;
fim-enquanto
retorne(x);
fim-função

(a) O que faz o algoritmo acima? (justifique)

(b) Calcule a complexidade de pior caso do algoritmo acima, contando o número de somas $x+y$. Descreva as entradas que levam o algoritmo ao pior caso.

10. Falso ou verdadeiro? (Justifique cuidadosamente.)

(a) Seja S uma lista linear sequencial com $n > 0$ elementos, n divisível por 5. Seja M a lista das medianas de

$$S[1 \cdots 5],\ S[6 \cdots 10], \cdots,\ S[n-4 \cdots n].$$

Então, a mediana de M é a mediana de S.

(b) Se as complexidades de melhor e pior caso de um algoritmo são ambas $\Theta(f(n))$, então a complexidade de caso médio também é $\Theta(f(n))$.

(c) Dados n blocos de base idêntica B mas com alturas distintas $h_1, \ldots, h_n$, onde cada h_i é um número real no intervalo $[0, 1]$, deseja-se colocá-los em caixas com base B e altura unitária, denotadas por C_1, C_2, etc. O seguinte algoritmo guloso resolve o problema de encontrar o menor número de caixas necessárias para armazenar todos os blocos: para cada $i = 1, \ldots, n$, coloque o bloco de altura h_i na caixa de menor índice que o possa comportar. (Portanto, o maior índice utilizado pelo algoritmo será o número de caixas usadas.)

11. Suponha uma lista de n números, representando os resultados de computação ("votos") de n processadores. Deseja-se decidir se há um *voto majoritário* e qual é este voto. (Dizemos que há um voto majoritário quando mais da metade dos processadores chegou ao mesmo resultado em sua computação.) Escreva um algoritmo eficiente para determinar se há um voto majoritário, baseado somente em comparações entre pares de votos, isto é, *não é permitido fazer contagem.* Calcule a complexidade do seu algoritmo. Você também pode interpretar este problema da seguinte forma: existem n esferas, cada qual com uma cor, e deseja-se saber se há uma cor majoritária entre elas.

12. Elabore um algoritmo de decomposição de um vetor S em três subvetores. Esse algoritmo recebe como entrada, além do vetor S, um valor piv pertencente a S, e os índices p e r, $1 \leq p \leq r$. O algoritmo deve rearrumar os elementos em $S[p \ldots r]$ e retornar dois índices i e j satisfazendo as seguintes propriedades:

    (a) se $p \leq k \leq i$, então $S[k] < piv$;
    (b) se $i < k \leq j$, então $S[k] = piv$;
    (c) se $j < k \leq r$, então $S[k] > piv$.

    A complexidade do seu algoritmo deve ser $O(n)$.

13. **(Os k Menores)** Suponhamos que em vez de selecionar o k-ésimo menor elemento de um conjunto com n elementos, estamos interessados em determinar os k menores elementos (sem nos preocuparmos com a ordem relativa entre eles). É possível realizar esta tarefa em tempo $O(n)$?

14. Elabore um algoritmo que, dado um vetor S com $n > 0$ elementos, retorna um vetor V, de tamanho n, com a seguinte propriedade: $V[i]$ é o número de ocorrências de $S[i]$ em S. A complexidade do seu algoritmo deve ser $O(n \log n)$.

15. **(Subsequência Monotônica Mais Longa)** Seja $S = s_1, s_2, \ldots, s_n$ uma sequência de números naturais. Uma *subsequência* de S é uma sequência da forma $S' = s_{i_1}, s_{i_2}, \ldots, s_{i_k}$, onde $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. O valor k é o *comprimento* de S'. A subsequência S' é dita *monotônica* se $s_{i_1} \leq s_{i_2} \leq \cdots \leq s_{i_k}$. Desenvolva um algoritmo de *divisão-e-conquista* para encontrar uma subsequência monotônica de S de comprimento máximo. Analise a sua complexidade.

16. **(Otimização de Transporte)** Na ilha de Tamoa, um guia de turismo necessita transportar K turistas de uma cidade origem s para uma cidade destino t. As cidades de Tamoa são representadas por vértices de um grafo conexo G. Cada aresta (u, w) de G representa uma estrada ligando u e w, na qual opera um serviço de ônibus que transporta no máximo $w(u, v)$ passageiros, tanto num sentido como no outro. Elabore um algoritmo, baseado numa estratégia gulosa, que determine o menor número de viagens de s a t que o guia necessita fazer para transportar todos os turistas.

17. Alguns projetores foram alocados para um congresso de Engenharia de Software na UFX, mas na "hora H" apenas um estava funcionando. Entre as n palestras previstas para o primeiro dia, os organizadores decidiram selecionar *o maior número possível* delas para serem apresentadas neste projetor, deixando as restantes para dias posteriores. Esta seleção deveria respeitar rigorosamente os horários marcados na programação. Na programação do evento, cada palestra p_i tem um horário de início s_i e um horário de término t_i $(1 \leq i \leq n)$. Um dos organizadores propôs o seguinte algoritmo guloso, onde A denota o conjunto de palestras selecionadas: (suponha que $t_1 \leq t_2 \leq \ldots \leq t_n$):

    $A \leftarrow \{p_1\}$
    $j \leftarrow 1$
    para $i = 2 \ldots n$ faça:
        se $s_i \geq t_j$ então $A \leftarrow A \cup \{p_i\}$
                $j \leftarrow i$

Este algoritmo funciona? Justifique sua resposta por A+B.

18. **(Sequenciamento de tarefas)** Seja $J = \{J_1, \ldots, J_n\}$ um conjunto de tarefas que devem ser executadas sequencialmente em um mesmo processador. Cada tarefa J_i consome uma unidade de tempo do processador e produz um lucro $r_i > 0$ caso seja concluída até o tempo limite t_i, $i = 1, \ldots, n$. Se J_i for concluída após t_i, nenhum lucro é obtido desta tarefa. O lucro total de uma sequência de execução das tarefas é a soma dos lucros obtidos pelas tarefas que foram concluídas até o tempo limite. Descreva um algoritmo que determina uma sequência de execução das tarefas em que o lucro total é maximizado. Prove que o algoritmo está correto. Qual a sua complexidade?

19. O Problema do Caixeiro Viajante consiste em um conjunto de n cidades e um custo não negativo C_{ij} associado a cada par de cidades i, j. O objetivo é determinar um caminho que, partindo da cidade 1, passe por todas as cidades (exatamente uma vez em cada cidade) e retorne à cidade 1, de modo que a soma dos custos no caminho seja mínima. Mediante um contra-exemplo, mostrar que o algoritmo guloso não resolve o Problema do Caixeiro Viajante.

20. Suponha que n tarefas devem ser executadas em duas máquinas A e B. Se a tarefa i for processada em A, serão consumidas a_i unidades de tempo; e se for processada na máquina B, b_i unidaddes de tempo. É possível que $a_i \geq b_i$, para algum i, e $a_j < b_j$ para algum $j \neq i$. Obtenha um algoritmo de programação dinâmica para determinar o mínimo tempo necessário para se executar todas as tarefas, supondo-se que as tarefas não podem ser divididas entre as duas máquinas (isto é, uma tarefa i ou é executada em A ou em B). Deve-se indicar como determinar a atribuição das tarefas a cada máquina na solução.

21. **(Triangularização ótima)** Escreva um algoritmo de programação dinâmica para o seguinte problema: dado um polígono convexo formado a partir de $n \geq 3$ vértices colocados sobre o plano cartesiano, encontrar uma triangularização deste polígono onde a soma dos perímetros dos triângulos obtidos seja a menor possível. Denote os vértices por $1, 2, \ldots, n$, e a distância entre os vértices i e j por $d(i, j)$. Qual a complexidade de seu algoritmo?

    Dica: Denote por $c(i, j)$ o custo mínimo de triangularização do polígono contendo os vértices $i, i+1, \ldots, j$, para $j \geq i+2$. Denote por $p(i, j, k)$ o perímetro do triângulo com vértices $i < j < k$, isto é, $p(i, j, k) = d(i, j) + d(i, k) + d(k, j)$. É claro que o objetivo do problema é calcular $c(1, n)$. Escreva uma fórmula recorrente para calcular $c(i, j)$, e deduza o algoritmo a partir desta fórmula.

22. **(Passeio Fechado de Cavalo)** Este problema consiste em encontrar um passeio *fechado* de cavalo sobre um tabuleiro $n \times n$ ($n \geq 6$). Isto é, a casa inicial do passeio deve coincidir com a casa final. Desenvolva um algoritmo para resolver o problema utilizando a técnica de *Backtracking*.

23. Uma matriz $n \times n$ contém como elementos $1, 2, \ldots, n^2 - 1$, além de um elemento branco, dispostos de forma arbitrária. A matriz pode ser modificada, mediante trocas sucessivas de posição entre o elemento branco e algum outro contíguo a ele, na mesma linha ou coluna. Descrever um método para transformar a matriz dada, mediante trocas de posição do elemento branco, em uma matriz em que os elementos apareçam ordenados nas linhas e colunas, com o elemento branco na posição inferior direita. Veja um exemplo abaixo. O problema sempre apresenta solução? Em caso negativo, quais são as condições necessárias e sufucientes para a existência de solução?

Configuração Inicial:

13	1	4	5
10	**B**	2	9
3	14	12	15
11	6	8	7

Configuração Final:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	**B**

24. Descrever um algoritmo de tempo polinomial para resolver o problema 2-SATISFABILIDADE. (As cláusulas têm no máximo dois literais cada uma.)

25. Uma *cobertura de ciclos* de um digrafo $D = (V, E)$ é um conjunto C de ciclos elementares, tal que cada vértice $v \in V$ pertence a exatamente um ciclo de C. Mostrar que o seguinte problema é NP-completo: Dado um digrafo $D(V, E)$ e um número $k \in Z^+$, D possui uma cobertura de ciclos formada por k ciclos ou menos?

26. Considere o problema de obtenção do percurso mínimo do caixeiro viajante, onde todas as distâncias são escolhidas dentre dois números inteiros a e b. Este problema está em P? É NP-completo?

27. Sejam A e B dois problemas tais que $A \in$ NP e $B \notin$ NP. Então existe uma transformação polinomial de B para A se e somente se P = NP. Certo ou errado?

28. Prove que o problema de determinar a árvore geradora de altura máxima de um grafo G é NP-difícil (e a de altura mínima?)

29. O problema CAMINHO MÍNIMO CONDICIONADO é assim definido: dados um grafo $G = (V, E)$ com pesos inteiros positivos nas arestas, vértices $v, t \in V$, um subconjunto $V' \subseteq V$ e um inteiro$k > 0$, a questão é: existe um caminho P entre s e t em G, tal que P contém todos os vértices de V' e a soma dos pesos das arestas de P é $\leq k$? Provar que esse problema é NP-completo.

30. Provar que o problema anterior permanece NP-completo quando G é substituído por um digrafo D, e torna-se polinomial quando D é um dígrafo acíclico.

31. O problema CAMINHO MÍNIMO CONDICIONADO permanece NP-completo quando os pesos das arestas do grafo são todos unitários?

32. Considere os seguintes problemas de decisão:

    - 2-SAT : Dada uma expressão booleana E na forma normal conjuntiva onde cada cláusula contém dois literais, pergunta-se: existe uma atribuição de verdade para E tal que cada claúsula contenha **pelo menos um** literal verdadeiro? **(Sabe-se que 2-SAT pertence a P.)**
    - 3-SAT$_{\overline{1}}$: Dada uma expressão boolena E na forma normal conjuntiva onde cada cláusula contém três literais, pergunta-se: existe uma atribuição de verdade para E tal que cada claúsula contenha **exatamente um** literal verdadeiro? **(Sabe-se que 3-SAT$_{\overline{1}}$ é NP-completo.)**
    - 3-SAT$_{\overline{2}}$: Dada uma expressão boolena E na forma normal conjuntiva onde cada cláusula contém três literais, pergunta-se: existe uma atribuição de verdade para E tal que cada claúsula contenha **exatamente dois** literais verdadeiros?
    - 3-SAT$_2$: Dada uma expressão boolena E na forma normal conjuntiva onde cada cláusula contém três literais, pergunta-se: existe uma atribuição de verdade para E tal que cada claúsula contenha **pelo menos dois** literais verdadeiros?

Determine a que classe de complexidade pertencem os problemas 3-SAT$_{\overline{2}}$ e 3-SAT$_2$.

33. Considere o problema de otimização COBERTURA DAS ARESTAS POR VÉRTICES: Dado um grafo G, encontre um subconjunto S de vértices de cardinalidade mínima tal que toda aresta de G tenha pelo menos um de seus extremos em S. O seguinte algoritmo é proposto para encontrar uma solução do problema: repita a operação "encontre uma aresta ainda não coberta e coloque seus dois extremos em S" até que todas as arestas estejam cobertas. Mostre que este algoritmo é 2-aproximativo.