

Nome:

Escolha apenas duas questões da parte de algoritmos (3 últimas) e duas da parte da Teoria da Computação (3 primeiras) para responder.

1. (2.5 pts) **Disserte** sobre os conjuntos de problemas NP, NP-Difíceis e NP-Completo. Desenhe diagramas de Venn mostrando a relação entre estes três conjuntos para o caso de $P = NP$ e $P \neq NP$.
2. (2.5 pts) **Disserte** sobre os seguintes aspectos da *Teoria da Computabilidade*:
 - a) qual é o seu principal foco de estudo
 - b) exemplos dos principais resultados (teoremas, teses, etc)
 - c) relevância para a Computação desses resultados principais
3. (2.5 pts) **Disserte** sobre os seguintes aspectos da *Teoria da Complexidade*:
 - a) qual é o seu principal foco de estudo
 - b) exemplos dos principais resultados (teoremas, teses, etc)
 - c) relevância para a Computação desses resultados principais
 - d) problemas em aberto
4. (2.5 pts) Dadas duas strings X e Y , com n e m caracteres cada uma respectivamente, o problema da substring máxima consiste em encontrar o tamanho da maior substring comum entre as duas strings. Por exemplo, suponha $X=CTGCGCATG$ e $Y=AGCATCAG$, o maior substring comum é GCAT o qual possui tamanho 4. A equação de recursão deste algoritmo é:

$$LCS(X_i, Y_j) = \begin{cases} 0 & \text{se } i=0 \text{ ou } j=0 \\ LCS(X_{i-1}, Y_{j-1}) + 1 & \text{se } x_i = y_j \\ \max(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)) & \text{se } x_i \neq y_j \end{cases}$$

onde X_i representa o substring de X com as primeiras i posições, enquanto que X_{i-1} representa o substring de X com as primeiras $i - 1$ posições. Definição similar se aplica ao Y .

Apresente o pseudocódigo de um algoritmo de programação dinâmica que resolve este problema. Qual a complexidade de tempo e espaço de pior caso do seu algoritmo?

5. (2.5 pts) Apresente o pseudo-código de um algoritmo de ordenação (sorting), baseado na técnica de divisão e conquista, que tenha a seguinte equação de recorrência: $T(n) = 3T(n/3) + n$. Qual a complexidade de tempo de pior caso de um algoritmo com esta equação de recorrência?

6. (2,5 pts) Haverá uma reunião entre diversos representantes de entidades do Rio Grande do Sul para discutir uma possível solução à crise do Estado. No entanto, há divergências entre diversos representantes. A reunião será numa mesa redonda e o organizador do evento quer alocar todos os convidados ao redor da mesa sem que duas pessoas com divergências sentem lado a lado. Com base numa consulta feita aos convidados, o organizador montou uma matriz simétrica de zeros e uns indicando em cada célula $M[i][j]$ se os convidados i e j possuem divergências (1) ou não (0). Este problema pode ser visto como o problema do ciclo hamiltoniano. Dada a matriz $M^{n \times n}$ de entrada, o organizador propôs o algoritmo abaixo o qual retorna a solução S ou $null$ caso uma solução não for encontrada. O algoritmo inicia com o candidato com o maior número de divergentes. A cada iteração, aloca ao lado do candidato anterior v o próximo candidato u com o maior número de divergentes, mas que não é divergente de v .

```
1 C := ContaDivergentes(M); // O vetor  $c^n$  contém em cada posição  $i$  o número de divergentes de  $i$ .
2 v := MoreDivergente(C); // entre todos os convidados,  $v$  é o que tem mais divergentes.
3 S[1] := v ;
4 C[v] := -1
5 for i=2 to n-1 do
6     max = -1; u:=-1;
7     for j=1 to n do
8         if (j ≠ v and M[v][j]==0 and C[j] > max) then
9             max := C[j];
10            u := j;
11        end
12    end
13    if u== -1 then
14        return(null); //problema sem solução
15    end
16    S[i] := u;
17    for j=1 to n do
18        if M[j][u] == 1 then
19            C[j] := C[j]-1;
20            C[u] := C[u] -1;
21            M[j][u] == M[u][j] == 0;
22        end
23    end
24    v := u;
25    C[v] := -1; //marque o candidato v para não ser mais selecionado
26 end
27 if M[u][S[1]] == 1 then
28     return(null);
29 end
30 return(S);
```

Sobre este problema responda:

- Qual a complexidade de pior caso de tempo do algoritmo? (Considere que a função random tenha custo $O(1)$)
- Discurse sobre a otimalidade do algoritmo. Ou seja, caso o problema tiver solução ótima de acordo com os dados passados, uma solução ótima obrigatoriamente será retornada? Prove que sim, ou dê um contra-exemplo.
OBS: Observe que neste problema toda solução factível é ótima.