

Exame de Qualificação  
PPGC – Instituto de Informática – UFRGS  
5 de agosto de 2017

Profs. Luciana Buriol e Álvaro Moreira

Nome: \_\_\_\_\_

**Algoritmos**

1. (1 pt) Avalie se a sentença abaixo é verdadeira ou falsa. Se for verdadeira, argumente a razão. Se não for, apresente um contra-exemplo.  
“Considere uma instância do *Stable Matching Problem* no qual existe um homem  $m$  e uma mulher  $w$  tal que  $m$  aparece como primeiro na lista de preferência de  $w$ , e  $w$  aparece em primeiro na lista de preferência de  $m$ . Então em cada *stable matching*  $S$  para esta instância, o par  $(m,w)$  pertence a  $S$ .”
2. (2 pts) O algoritmo abaixo recebe como entrada um grafo  $G = (V, E)$  onde  $V$  é o conjunto de nós,  $E$  é o conjunto de arestas, e  $|V|=n$ ,  $|E|=m$ . O algoritmo é chamado passando o grafo  $G$  e um nó  $u$  qualquer do grafo. Responda as seguintes questões em relação ao algoritmo:
  - a) Suponha que o grafo  $G$  seja representado como uma lista de adjacências, qual a complexidade de pior caso do algoritmo?
  - b) Suponha que o grafo  $G$  seja representado como uma matriz de adjacência, qual a complexidade de pior caso do algoritmo?
  - c) Analise a complexidade de pior caso de b) e c) quando o algoritmo é aplicado em grafos do tipo árvore (neste caso  $m = n - 1$ )?

---

**Algorithm 1**  $ALG(G,u)$

---

```
1: Mark  $u$  as "Explored"
2: for all  $(u,v)$  incident to  $u$  do
3:   if  $v$  is not marked Explored then
4:     Recursively invoke  $ALG(G,v)$ 
5:   end if
6: end for
```

---

3. (2) Dados dois strings  $A$  e  $B$  de tamanhos  $m$  e  $n$ , respectivamente, apresente o pseudo-código ou a equação de recorrência de um algoritmo de programação dinâmica que calcula o menor número de operações necessárias para transformar  $A$  em  $B$ . As operações permitidas são de inserção de um caracter, remoção de um caracter, ou troca entre caracteres. Analise a complexidade de pior caso do seu algoritmo.  
Por exemplo, para  $A=SONHAR$  e  $B=VOAR$ , o menor número de operações é 3: troca  $S/V$ , remoção de  $N$ , remoção de  $H$ .

## Teoria

1. (2,5 pts) Disserte sobre os seguintes tópicos abaixo. Deixe bem claro o significado e a importância de cada um para a Ciência da Computação:

- a) (0,5) Tese de Church-Turing
- b) (0,5) Problema da Parada
- c) (0,5) Teorema de Rice
- d) (0,5) Teorema de Cook-Levin
- e) (0,5) Questão "P = NP?"

2. (2,5 pts)

- a) (0,5) Dê uma definição de *redução de tempo polinomial*.
- b) (0,5) Dê uma definição da *classe de complexidade NP*.
- c) (0,5) Usando os conceitos (a) e (b) acima dê uma definição de problema *NP-Hard*
- d) (0,5) Usando os conceitos acima, dê uma definição de problema *NP-Completo*
- e) (0,5) Seja Double-SAT o conjunto de fórmulas booleanas  $\phi$  em formato CNF (forma normal conjuntiva) tais que  $\phi$  é satisfeita por pelo menos duas atribuições de valores verdade às suas variáveis. Prove que o problema Double-SAT é NP-Completo. Nesta prova deixe bem claro que Double-SAT satisfaz todos os itens da sua definição de problema NP-Completo dada no item (d) acima. Na sua resposta você pode também usar outros resultados conhecidos sobre a complexidade do problema de satisfatibilidade Booleana.

Obs.: uma fórmula está no formato CNF se ela é uma conjunção de disjunções de literais. A seguinte fórmula por exemplo, está em formato CNF:

$$(x_1 \vee x_2) \wedge (x_3 \vee \neg x_1 \vee \neg x_4) \wedge (\neg x_2 \vee x_5)$$