Exame de Qualificação 2016-1.N1 Data:08/03/2016
Professoras: Leila Ribeiro e Luciana Buriol

Cartão: **Nome:**

Dicas gerais:

- Leia todas as questões antes de começar a prova e pergunte em caso de dúvidas.
- Responda a cada questão, ainda que a resposta não esteja completa.
- Justifique as suas respostas.

OBS: Resolva apenas duas questões da parte de algoritmos.

1. (2.5 pts) **Problema da mochila ilimitado**: dada uma mochila com capacidade de peso C, e um conjunto de n tipos de itens, cada tipo de item $i = 1, \cdots n$ com um peso $w_i \in N$ e um valor $v_i \in R^+$. O problema consiste em definir a soma máxima do valor dos itens que podem ser inseridos na mochila sem extrapolar sua capacidade, considerando que infinitas cópias de cada tipo de objeto estão disponíveis. Apresente o pseudocódigo de um algoritmo de programação dinâmica que resolve o problema, ou a equação de recursão correspondente.
OBS: o algoritmo clássico de PD para resolver este problema requer memória $O(C)$.

$$OPT[c] = \begin{cases} OPT[c-1] & \texttt{if } w_i > c \\ max(OPT[c-1], OPT[c-w_i] + v_i) & \texttt{if } w_i \leq c \end{cases}$$

2. (2.5 pts) Apresente um algoritmo de divisão e conquista que a cada execução da recursão faça apenas uma chamada recursiva, e o tamanho de um problema do nível $i+1$ seja $\frac{1}{3}$ do tamanho do problema no nível i. Todas as operações dentro de cada chamada recursiva são constantes. Apresente a equação de recorrência do algoritmo, bem como analise sua complexidade de tempo. Qual problema o seu algoritmo resolve?

   Divisão ternária (vem vez de binária)
   T(n) = T(n/3) + O(1)
   $T(n) \in O(n)$.

3. (2.5 pts) O pseudocódigo 1 representa o algoritmo A^* que calcula o caminho mínimo entre os vértices $s \in V$ (source) e $d \in V$ (destino) do grafo direcionado G=(V,A), onde V representa o conjunto de vértices e A representa o conjunto de arcos do grafo. Os arcos possuem custos $w_a \in R^+$ não negativos. A^* seleciona o caminho mínimo considerando a função f() de forma que: $f(v) = g(v) + h(v)$, onde v é um vértice no caminho, g(v) é

o custo do caminho do vértice s a v, h(v) é um valor heurístico que estima o custo de v até d. O algoritmo apresentado considera que a heurística h() é admissível, ou seja, nunca superestima o custo real de v até o destino, e h() é monotônica, isto é, $h(v) \leq w(v, y) + h(y)$ para todo $a = (v, y) \in A$ de forma que cada vértice é processado uma única vez. O conjunto Open é implementado como uma fila de prioridades (heap binário), enquanto Closed é um vetor binário indicando se cada vértice está ou não em Closed. A função h() pode ser calculada em tempo constante. As funções extractMin(S), add(u,S), e update(u,S) significam remover o vértice com chave de valor mínimo da estrutura S, adicionar o vértice u na estrutura S, e atualizar a estrutura S de acordo com o valor da chave de u, respectivamente. Todas operações em Open consideram o valor de f() como chave.

Algorithm 1 Calcula o custo do caminho mínimo entre os vértices s e d.

```
function A*(G = (V, A), s, d ∈ V)
    Closed ← ∅;
    Open ← {s}
    for each vertice u ∈ V do
        g_u ← ∞;
        f_u ← ∞;
    end for
    g_s ← 0;
    f_s ← h(s);
    while Open ≠ ∅ do
        u ← extractMin(Open);
        if u == d then
            return(f(u));
        end if
        add(u,Closed);
        for each v neighbour of u do
            if v in Open then
                if g(u) + w(u, v) < g(v) then                ▷
                    g(v) ← g(u) + w(u, v);
                    f(v) ← g(v) + h(v);
                    update(u,Open);
                end if
            else if v ∉ Closed then                        ▷ Novo nó
                g(v) ← g(u) + w(u, v);
                f(v) ← g(v) + h(v);
                add(v,Open);
            end if
        end for
    end while
end function
```

Analise a complexidade de tempo do algoritmo 1 segundo as condições abaixo:

(a) O grafo é representado como uma lista de adjacência;
$O(mlogn)$ considerando grafo conexo.

(b) O grafo é representado como uma matriz de adjacência.
$O(n^2 logn)$

Exame de Qualificação:
Questões sobre Teoria da Computação
08/03/2016

Nome:

Dicas gerais: Leia todas as questões antes de começar; sempre justifique a sua resposta. A avaliação levará em consideração a abrangência e a profundidade demonstradas nas soluções apresentadas.

Questão 1 (2.5 pontos) Na área de Computação, um conceito fundamental é o de *Máquina Universal.*

a) Defina a Máquina de Turing Universal, explicando intuitivamente a definição e discutindo os requisitos para que esta definição seja possível (por exemplo, o fato do alfabeto de uma Máquina de Turing ser finito é importante para a definição da a Máquina Universal?);

b) Explique a relevância desta definição para a Ciência da Computação;

c) Existem conceitos análogos ao de Máquina de Turing Universal em outros modelos de computação, por exemplo em Funções Recursivas Parciais? Explique.

Questão 2 (2.5 pontos) Disserte sobre *Complexidade de Problemas*, incluindo os seguintes tópicos, entre outros que você julgue importantes (identifique claramente no seu texto onde estes tópicos estão desenvolvidos):

a) Classes de complexidade;

b) Redução de problemas (defina e dê exemplos);

c) Teorema de Cook-Levin (explique intuitivamente o que diz este teorema, bem como sua importância);

d) A questão $\mathbf{P} = \mathbf{NP}$? (o que significa e sua relevância para a Ciência da Computação).