

Rio de Janeiro, 13 de Setembro de 2007.

PROVA 1 DE ANÁLISE DE ALGORITMOS

PROFESSOR: EDUARDO SANY LABER

DURAÇÃO: 2 HORAS

1. (2.0pt) Seja $G = (V, E)$ um grafo direcionado.

a) Explique como seria um algoritmo para encontrar um par de vértices $u, v \in V$ tal que não existe caminho de u a v em G e nem caminho de v a u em G . Caso tal par não exista, o caminho deve indicar "NÃO EXISTE".

b) Análise a complexidade do algoritmo proposto.

2. (2.0pt) Explique o que é uma árvore AVL indicando suas possíveis vantagens e desvantagens em relação a uma árvore binária de busca tradicional.

3 (2.0pt). Considere o seguinte algoritmo que recebe um número inteiro N como entrada e determina se N é primo ou composto.

```
I=2
While I * I ≤ N faça
    If N é múltiplo de I then
        Return N é composto
    I:=I+1
Return N é primo
```

a) Faça uma análise assintótica da complexidade de tempo deste algoritmo?

b) Qual o tamanho da entrada em função de N ?

c) Este algoritmo é polinomial no tamanho da entrada ? Por que ?

4.(2.0pt) Análise em função de n a complexidade de tempo do procedimento abaixo?

Para i variando de 1 a n faça

$t \leftarrow n$

$q \leftarrow 1$

Enquanto $q \leq t$

$q \leftarrow 2q$

Fim Enquanto

Fim Para

5. (2.0pt) Considere o pseudo-código da busca em largura abaixo.

(a) Modifique o pseudo-código para que este passe a calcular, para cada nó $v \in V$, a distância de v ao nó de origem da busca s

(b) Modifique o pseudo-código para que este determine se o grafo de entrada é bipartido ou não. Assuma neste item que já está disponível um vetor $Dist$, onde $Dist(v)$ indica a distância do vértice v até a origem da busca s .

BFS	
Procedure BFS(G,s)	
1.	Marque s como visitado
5.	ENQUEUE(Q,s)
9.	while $Q \neq \emptyset$
10.	$u \leftarrow$ DEQUEUE(Q)
11.	For each $v \in Adj[u]$
12.	if v não visitado then
14.	Marque v como visitado
16.	ENQUEUE(Q,v)
20.	End For
30.	End While

Figura 1: Pseudo-Código de uma BFS