Rio de Janeiro, 24 de Abril de 2007.
PROVA 1 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER

1. Obtenha como função de n a melhor análise de complexidade possível para os dois pseudo-códigos apresentados abaixo.

Pseudo-Código-1
    t $\leftarrow$ 0
    Cont $\leftarrow$ 1
    Para i=1 até n
        Cont $\leftarrow$ 2*COnt
    Fim Para

    Enquanto Cont $\geq$ 1
        Cont $\leftarrow$ Cont/2
        Para $j = 1$ a n
            $t++$
        Fim Para
    Fim Enquanto

---

2. Seja um grafo não direcionado $G = (V, E)$. Uma cobertura para G é um conjunto de vértices $C \subseteq V$ tal que toda aresta de E tem pelo menos uma de suas extremidades em C. Uma cobertura C é dita ótima se não existe outra cobertura C' tal que $|C'| < |C|$. Considere o seguinte algoritmo guloso apresentado na Figura 1.

$C \leftarrow \emptyset$;
Enquanto G tem alguma aresta
    $v \leftarrow$ vértice de G com maior grau
    $C \leftarrow C \cup v$
    Remova v de G [1]
Fim Enquanto
Devolva C

Figura 1: Cobertura

a) Exiba um exemplo mostrando que este algoritmo nem sempre devolve a cobertura ótima.

b) Explique como implementar o algoritmo acima em $O(|E| \log |V|)$. Assuma que o grafo é dado por uma lista de adjacências.

3.(2.0pt) Seja $G = (V, E)$ um grafo conexo e não direcionado. Uma ponte em G é uma aresta $e \in E$ tal que se removermos e de G o grafo fica desconectado.

a) Mostre que se a aresta uv não é uma ponte então existe um ciclo em G que contem os vértices u e v.

b) Descreva com palavras como seria um algoritmo para contar o número de pontes do grafo. Qual a complexidade do algoritmo proposto.

4. No código abaixo Q é um tipo abstrato de dados que suporta as seguintes operações

- Insira(Q,j). Esta operação insere o inteiro j em Q
- Remove-Mínimo(Q). Esta operação remove o menor elemento de Q

e $rand(j)$ retorna um inteiro aleatóreo entre 1 e j. Assuma que $Rand()$ gasta $O(1)$.

Para i variando de 1 a n^2 faça.

Insira(Q, rand($10n^2$))

Para i variando de 1 a n^2 faça.

Remova-Mínimo(Q)

a) Qual a complexidade do pseudo-código acima se Q é implementado como uma lista encadeada

b) Qual a complexidade do pseudo-código acima se Q é implementado como um heap binário

5. (2.0pt) Seja $G = (V, E)$ um grafo direcionado. Assuma que cada vértice v do grafo tem uma importância $Imp(v)$ que pode ser obtida em $O(1)$. A *qualidade* de um nó u é dada pela fórmula $\sum_{v \in V} d(u, v) Imp(v)$, onde $d(u, v)$ ' é a distância de u a v no grafo G.

Modifique o pseudo-código da busca em largura abaixo para que este determine o vérice de V com maior qualidade. Análise a complexidade do algoritmo desenvolvido

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 2: Pseudo-Código de uma BFS