Rio de Janeiro, 4 de Abril de 2011.
PROVA 1 DE ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 1:50h

1. (2.0) Considere o pseudo-código a seguir

```
Proc(n)
    Se n =1
        Return
    Fim Se
    Para i:=1 até n-1
        Para j:=i+1 até n
            t ← 0
        Fim para
    Fim para
    Proc(n-1)
Fim Proc
```

a) Seja $T(n)$ a complexidade de pior caso do procedimento acima. Ache uma equação de recorrência para T(n).

b) Encontre uma função $f(n)$ tal que $T(n) = \Theta(f(n))$.

2. (1.5pt) Seja

$$T(n) = \sum_{i=1}^{n} \frac{n \log n \times i^2}{2^i}.$$

Encontre uma função $f(n)$ tal que $T(n) = \Theta(f(n))$.

3. (2.5pt) Seja $S = \{a_1, \ldots, a_n\}$ um conjunto de n números reais distintos. Considere o problema $\mathcal{P}$ de determinar se existem três números distintos em S cuja soma é 0

a) Seja $T(n)$ a complexidade de pior caso do algoritmo abaixo para resolver $\mathcal{P}$. Encontre $f(n)$ tal que $T(n) = \Theta(f(n))$.

Para i=1,...,n-2
  Para j=i+1,...,n-1
    Para k=j+1,...,n
      Se a_i+a_j+a_k =0
        Return SIM
Return NÃO

b) Projete um algoritmo mais eficiente do que o algoritmo acima em termos de complexidade assintótica. Não é necessário apresentar o pseudo-código mas sim explicar com clareza os passos que o algoritmo deve realizar e explicar a complexidade. Quanto mais eficiente o algoritmo maior a pontuação.

4 (2.0pt). Considere o problema $\mathcal{Q}$ definido da seguinte forma: *Entrada*: Inteiro $N \geq 3$; *Saída*: SIM se N é composto; NÃO, caso contrário.

a) Qual é o *tamanho da entrada* deste problema em função de N.

b) Determine a função complexidade de tempo do algoritmo acima para resolver $\mathcal{Q}$. Este algoritmo é polinomial ? Por que?

```
I ← 2
Enquanto I ∗ I ≤ N faça
        If N é múltiplo de I then
                Return SIM, N é composto
        I ← I + 1
Return NÃO, N é primo
```

5. (2.0pt) Seja S um conjunto de n inteiros no conjunto $\{1, ..., U\}$. Considere o seguinte procedimento

Para i=1,...,n^2

  Remova o menor elemento de S

  $x \leftarrow Rand(1, U)$ [1].

  Insira x em S.

Fim Para

a) Analise a complexidade do algoritmo acima quando S esta armazenado como um heap binário. Note que após cada inserção e remoção devemos restaurar a ordenação do heap.

b) Assuma agora que $1 \leq U \leq 5$. Como poderiamos armazenar S de modo a melhorar a complexidade do procedimento? Qual a complexidade obtida?

6. (2.0pt) Seja S um conjunto de n numeros reais. Explique como seria um algoritmo eficiente para encontrar os $\sqrt{n}$ menores números do conjunto S e analise sua complexidade. Quanto mais eficiente o algoritmo maior a pontuação.

[1]RAND(1,U) retorna em tempo constante um inteiro aleatório entre 1 e U