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 $T(n) = \Theta(f(n))$.

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 Q. Este algoritmo é polinomial ? Por que? (Assuma que cada teste "N é multiplo de I" é O(1))

```
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
```

3. (2.0pt) Seja S um conjunto de n elementos, onde cada elemento $s \in S$ tem uma prioridade $p(s)$ pertencente ao conjunto dos U menores inteiros positivos. Considere o seguinte procedimento

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

$x \leftarrow Rand(1, U)$

Modifique a prioridade do elemento de S com menor prioridade para x (*)

Fim Para

Assuma que a função $RAND(1, U)$ retorna em tempo constante um inteiro aleatório entre 1 e U. Além disso, caso haja mais de um elemento com prioridade mínima na linha (*), qualquer um pode ser escolhido.

a) Analise a complexidade do algoritmo acima quando S esta armazenado como um heap binário de mínimo. Note que após cada mudança de prioridade 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? Explique como seria o procedimento para modificar a prioridade do menor elemento para a nova estrutura de armazenamento proposta. Qual seria a complexidade obtida?

5. (2.0pt) Seja $G = (V, E)$ um grafo não direcionado e seja uv uma aresta de G. Explique com palavras como seria um algoritmo polinomial para determinar se existe um ciclo no grafo que contém a aresta uv. Qual a complexidade deste procedimento?