Rio de Janeiro, 01 de Abril de 2009.

PROVA 1 DE ANÁLISE DE ALGORITMOS

PROFESSOR: EDUARDO SANY LABER

DURAÇÃO: 1:50h

1. (3.opt) Considere o seguinte procedimento para manipular uma lista A.

Proc(A)
Se $|A|$=1
    Return
Senão
    Embaralhe(A)
    Ordene(A)
    Divida A em duas listas A_e e A_d, cada uma delas com o mesmo tamanho.
    Proc(A_e)
    Proc(A_d)
Fim Proc

a) Seja $T(n)$ o número de operações realizadas por Proc para a pior instância de tamanho n. Assuma que o embaralhamento consuma $2n$ operações e que a divisão em A_e e A_d consuma $O(1)$ operações. Além disso, assuma que *ordene*(A) execute $g(n)$ operações para pior instância de tamanho n. Escreva uma equação de recorrência para $T(n)$.

b) Resolva a recorrência para o caso em que Ordena é o QuickSort.

c) Resolva a recorrência para o caso em que Ordena é o HeapSort.

2. (3.0pt) Seja $S = \{a_1, \ldots, a_n\}$ um conjunto de n números naturais distintos e um inteiro x. Considere o problema $\mathcal{P}$ de determinar se existem três números naturais distintos em S cuja soma é x

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 =x
        Return SIM
Return NÃO

b) Projete um algoritmo com complexidade $O(n^2 \log n)$ para resolver o problema $\mathcal{P}$. Não é necessário apresentar o pseudo-código mas sim explicar com clareza os passos que o algoritmo deve realizar e explicar a complexidade.

3 (2.0pt). Considere o problema P 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.

```
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
```

b)) Determine a função complexidade de tempo do algoritmo acima para resolver P. Este algoritmo é polinomial ? Por que?

4. (3.0pt) 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?

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