UNIVERSIDADE DE SÃO PAULO - EACH

SIN5013 PROVA 1

PROFESSORA: Karina Valdivia Delgado
Programa de Pós-graduação em Sistemas de Informação
Nome:
Número USP:

---

1. Suponha que T(n) = O(f(n)), mostre que T(n) + f(n) = O(f(n)). **(2 pontos)**

Desejamos demonstrar que T(n)+f(n) <= c f (n) para todo n>n0

T(n) <= c1 f(n) para todo n>=n01 (0,25)

T(n) + f(n) <= c1 f(n) + f(n) para todo n>=n01 somando f(n) em ambos lados (0,75)

T(n) + f(n) <= f(n) (c1 + 1) para todo n>=n01 colocando em evidência f(n)

c= c1+1 n0=n01
(0,5) (0,5)

2. A ordenação por inserção pode ser expressa sob a forma de um procedimento recursivo como a seguir. Para ordenar A[1...n], ordenamos recursivamente A[1...n-1] e depois inserimos A[n] no arranjo ordenado A[1...n-1]. Para uma entrada de tamanho n, seja T(n) o consumo de tempo dessa versão recursiva da ordenação por inserção considerando o pior dos casos. **Encontre a equação de recorrênci**a, isto é deduza do algoritmo uma recorrência que defina T(n). Não é necessário resolver a equação, apenas identificá-la. **(2 pontos)**

T(n) = T(n-1)+n-1
T(n-1)+O(n)
T(n-1) +4n

(1,0) (1,0)

3. Seja a equação de recorrência:

$T(1)=8$

$T(n)= 4T(n/2) +3n$, para $n>1$ e n potência de 2.

Use **indução matemática** para mostrar que **$T(n) >= 5n^2$**
(2 pontos)

Caso base:

$T(n) >= 5n^2$

$T(1) >= 5 * 1 = 5$

$8>=5$ ok

(0,5)

Passo indutivo:

$T(n)= 4T(n/2) +3n$

indução forte hi $T(n/2) >= 5 (n/2)\text{^}2$

$T (n) = 4T(n/2) +3n \overset{hi}{>=} 4 [5 (n/2)\text{^}2] + 3n$ (0,5)

$= 5 n\text{^}2 +3n$ (0,5)

$>= 5n\text{^}2$ (0,5)

4. Uma fila de prioridade é um tipo abstrato de dados que mantém um conjunto S de elementos, cada qual com uma prioridade associada e que suporta as seguintes operações:

- Maximum(S): retorna o elemento de S que possui a maior prioridade
- Extract-max(S): remove e devolve o elemento de S que possui a maior prioridade
- Insert(S, x, prior): insere o elemento x que tem uma prioridade prior no conjunto S

A estrutura de dados max-heap, por sua vez, é uma estrutura de dados que tem a seguinte propriedade: o conteúdo de um nó é maior ou igual que o conteúdo dos nós na subárvore enraizada nele. Um heap pode ser visualizado como uma árvore binária. A árvore é completamente preenchida em todos os níveis, exceto possivelmente o último nível em que as folhas estão o mais à esquerda possível.

A fila de prioridade pode ser implementada usando um max-heap representado por um vetor **A** de tamanho **m**. Elabore o pseudocódigo do método Extract-max que usa um max-heap. **(2 pontos)**

maior = A[1] **(0,5)**

A[1] <=> A[m] (0,5)

m<- m-1 **(0,5)**

max-heapify (A , m , 1) **(0,5) = corrige-descendo**

devolve maior

5. Um aluno diz que descobriu um novo **algoritmo de pesquisa (busca)** super eficiente para uma entrada de tamanho n com implementação recursiva de complexidade **T(n) = 3T(n/3) + n para n>1** e potência de 3 e com **custo constante igual a 8 para n=1**. Resolva a equação de recorrência de maneira exata usando o método da iteração ou árvore de recorrência e indique se você recomendaria a implementação desse algoritmo ou não? **(2 pontos)**

T(1)=8

T(n) = 3T(n/3) + n

T(n) = 3 [3 T[n/9]+n/3] +n = 3 ^2 T[n/9] + n +n = 3 ^2 T[n/9] + 2n

T(n) = 3 ^2 [3 T (n/27) +n/9] + 2n = 3^3 T[n/27] + n +2n= 3^3 T[n/27] + 3n 0,5

iteração i
T(n) = 3^i T (n/3^i) + i n 0,5

n/3^i =1 n= 3^i log_3 n = i

= n T(1) + n log_3 n 0,5

= 8 n + n log_3 n

Theta(n log n)

Não recomendo pois existem algoritmos com consumo theta(n) ou theta(log n) correspondentes a busca linear e binária, respectivamente. 0,5