UNIVERSIDADE DE SÃO PAULO - EACH

SIN5013 PRIMEIRA PROVA

1. (2.5 pontos) *Suponha que T(n) = Ω (f(n)). Demonstre que f(n) = O (T(n)). Coloque no final o valor de c e n_0 que foram encontrados para fazer essa demonstração.*

1) $T(n)$ é $\Omega(f(n)) \therefore \exists\, c \wedge n$ que satisfaz:

(1) $T(n) \geq c_1 . f(n) \quad \forall\, n \geq n_0$

Para provar que $f(n) = O(T(n))$, irei demonstrar a validade de (2):

(2) $f(n) \leq c_2 . T(n)$

Irei reescrever 1 de maneira a inverter os Termos da direita e esquerda:

(3) $c_1 . f(n) \leq T(n)$

Multiplicando os dois lados de (3) por $\frac{1}{c_1}$, temos:

(4) $f(n) \leq \frac{1}{c_1} . T(n)$

Note que 4 é uma expresão de mesma forma que 2. Temos que

$c_2 = \frac{1}{c_1} \quad n_0 = n_{01}$

$\forall$ *n >= n_{01}*

definição=0.5

se partiu da conclusão e não da premissa (suposição) não foi considerada pontuação nesta parte

manipulação matemática: 1.5

resposta=0.5

2. (2.5 pontos) *Seja a equação de recorrência:*

T(1)=8

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

***Use indução matemática** para mostrar que $T(n) = 8n + n\log_3 n$.*

***OBS**: Não esqueça de demonstrar também o caso base.*

TEM QUE SER INDUÇÃO FORTE

não pode demonstrar com indução fraca: faz a suposição que vale para k e demonstra que vale para k+1

Solução 1:

02 - Seja $T(1) = 8$ $T(n) = 3 \cdot T(n/3) + n$, p/ n>1 e pot. 3 Demonstrar que $T(n) = 8n + n\log_3 n$ Usando indução finita forte: i) Passo base: $n = 1 \rightarrow T(1) = 8$ $T(1) = 8 \cdot 1 + 1 \cdot \log_3 1 = 8 + 0 \rightarrow \boxed{T(1) = 8}$ ✓ A condição se satisfaz.	***CASO BASE:0.25***
ii) Passo indutivo: Assumimos que a propriedade é válida p/ n/3 e queremos provar que também é válida p/ n. Então, $T(n/3) = 8(n/3) + (n/3) \cdot \log_3 n/3$ $T(n/3) = (8/3)n + n/3 \cdot \log_3 n - n/3 \log_3 3$ $T(n/3) = 8/3 \cdot n + n/3 \cdot \log_3 n - n/3$	***manipulação matemática***
Agora, substituindo na equação de recorrência: $T(n) = 3 \cdot \left[8/3\, n + \frac{n}{3}\log_3 n - \frac{n}{3}\right] + n$	***Substituição da H.I. =1.25***
$T(n) = 8n + n\log_3 n - n + n$ $\boxed{T(n) = 8n + n\log_3 n}$ Lo que é exatamente o que queremos provar!	***Manipulação matemática: 1.0***

Solução 2:

Paso base: 0.25	② base: $n = 1$ $T(1) = 8$ $\Rightarrow 8.1 + 1.\log_3 1 = 8 + 1.0 = 8$
Paso indutivo: queremos provar que: equação de recorrência: substituindo a H.I. na equação de recorrência 1.25 manipulação matemática: 1 que é exatamente o que desejavamos provar	Indutivo: H.I $\Rightarrow k \in \mathbb{N} \mid T(k) = 8.k + k.\log_3 k$ p/ k como potencia de 3 $T(3k) = 8.3k + 3k\log_3 3k$ $T(3k) = 3T(3k/3) + 3k \Rightarrow T(3k) = 3T(k) + 3k$ $T(3k) = 3(8k + k.\log_3 k) + 3k$ $T(3k) = 8.3k + 3k\log_3 k + 3k$ $T(3k) = 8.3k + 3k(\log_3 k + 1)$ $T(3k) = 8.3k + 3k(\log_3 k + \log_3 3)$ $T(3k) = 8.3k + 3k\log_3 3k$ //

3. (2.5 pontos) *Resolva a seguinte equação de recorrência* ***de maneira exata*** *usando o* ***método da iteração*** *passo a passo:*

T (1) = 3

T (n) = 2 T(n/2) + n para n>=2 e n potência de 2.

OBS: Não precisa usar indução para verificar o resultado.

(I) Método da iteração:

iteração	Original	Transf.
1	$2T(n/2)+n$	
2	$2.[2.T(n/4)+n/2]+n$	$2^2T(n/2^2)+2n$
3	$2.[2.[2T(n/8)+n/4]+n/2]+n$	$2^3T(n/2^3)+3n$
⋮	⋮	⋮
i		$2^iT(n/2^i)+in$

(II) Quando para (Caso base)?

$T(1) = T(n/2^i)$

$1 = n/2^i$

$2^i = n$

$i \log 2 = \log n$

$\boxed{i = \log n}$

(III) Por fim, temos que substituindo na equação da iésima iteração, temos

$T(n) = nT(1) + n.\log n$

$T(n) = n.3 + n\log n$

$\boxed{T(n) = 3n + n\log n}$

iteração 1,2 e 3 :1ponto

iteração i + quando para: 1 ponto

a substituição do i na formula: 0.5

4. (2.5 pontos) *Considere o código apresentado a seguir.*

```
/* n uma potencia de 3 */
void Sort2 (int* A, int i, int j)
{
  if ( i < j )
  {
    m = ( (j - i) + 1 )/3;
    Sort2( A , i , i + m - 1 );
    Sort2( A , i + m , i + 2*m - 1 );
    Sort2( A , i + 2*m , j );
    Merge( A , i , i + m , i + 2*m , j );
    /* Merge intercala A[i..(i+m-1)], A[(i+m)..(i+2m-1) e A[i
       +2m..j] em A[i..j] a um custo ( (5n/3) -2 ) */
  }
}
```

Seja G(n) o consumo de tempo do algoritmo "Sort2" em que A é um vetor. Considerando que ***n=j-i+1*** *e que n é uma potência de 3, encontre a equação de recorrência, isto é deduza do algoritmo a recorrência que define G(n). Não é necessário resolver a equação, apenas identificá-la. O algoritmo "Merge" intercala os 3 subvetores e tem um custo de 5n/3-2.*
OBS: *Não esqueça do caso base.*

```
1 Sort2 (A, i, j)
2     Se i < j então                          Θ(1)
3         m ← ((j-i)+1)/3                     Θ(1)
4         Sort2(A, i, i+m-1)                  T(n/3)
5         Sort2(A, i+m, i+2*m-1)              T(n/3)
6         Sort2(A, i+2*m, j)                  T(n/3)
7         Merge(A, i, i+m, i+2*m, j)          5n/3 + 2
```

- Passo base

O vetor só possuí 1 elemento, portanto, somente a linha 2 é executada.

$G(1) = 1$

- Para $n \geq 3$ e $n = 3^i$

$G(n) = 3T(n/3) + \frac{5}{3}n + 4$ para $n \geq 3$ e $n = 3^i$

$G(1) = 1$

$G(n) = 3T(n/3) + \frac{5n}{3} + 4$ para $n \geq 3$ e $n = 3^i$

caso base: 0.5
3T(n/3) 1
5n/3+2 1

Caros alunos de SIN5013,

A prova começará às 14hs. A seguir alguns comentários:

1. São 4 perguntas e o tempo estimado para resolver a prova é 1 hora e 15 minutos. Porém, poderão submeter a prova até às 15:59. Recomendo baixar o pdf da prova.

2. Cada pergunta deve ser resolvida em uma página separada e na mão. Após terminar a prova devem submeter uma foto de cada pergunta em formato pdf, png ou jpg no e-disciplinas.

3. Todos os alunos devem estar com a câmera aberta.

4. Não é permitida consulta de nenhum material no computador, no celular ou qualquer outro tipo de aparelho. Apenas podem consultar material impresso e o pdf da prova que está disponível no e-disciplinas.

5. Até amanhã darei um retorno se a prova 2 será online ou presencial.

6. O aluno deve estar preparado com outras opções de internet caso a rede do local fique com problemas.

Desejo uma boa prova para vocês.

Karina