

Disciplina: Projeto e Análise de Algoritmos
Professor: Edleno Silva de Moura
Primeiro Trabalho Prático

Entregar as partes impressas para correção.
ENTREGA 04/09/2019

1 – Para cada afirmação abaixo indique se a mesma é verdadeira ou falsa. Justifique suas respostas (10%)

- A) Se $f(n) = 2^{1,7n}$ então $f(n)$ é $O(2^{1,5n})$
- B) Se $f(n) = 1,2n$ então $f(n)$ é $\Omega(3n)$
- C) Se $f(n) = n \log(n)$ então $f(n) = O(n^2)$ e $f(n) = \Omega(n)$ e $f(n) = \Theta(n \log(n))$

2 – Avalie as seguintes equações de recorrência apresentando a função de custo, sem deixar somatórios e incluindo eventuais constantes. Faça as suposições que forem necessárias para evitar chamadas com valores de n que não sejam inteiros (20%)

$$A) t(n) = \begin{cases} n \leq 0 & 0 \\ \text{senão} & 2t(n-4) + 1 \end{cases}$$

$$B) t(n) = \begin{cases} n > 1 & 4t(n/2) + 1 \\ n = 1 & 1 \end{cases}$$

3 – Apresente o número de vezes que a operação `faz_algo` é executada em função de n : (20%)

```
a) VERIFICA_ALGO(int n) {  
    INT x,y,z;  
  
    FOR(x=0; x<n; x++) {  
        FOR(y=0; y<x; y++) {  
            FOR(z=y; z<x; z++) { faz_algo; }  
        }  
    }  
}
```

```
b) VERIFICA_ALGO(int n) {
```

```

    INT x,y,z;

    FOR(x=0; x< n; x++) {
        FOR(y=0; y<n; y++) {
            FOR(z=y; z<n; z++) {  faz_algo;    }
        }
    }
}

```

```

c) VERIFICA_ALGO(int n) {
    INT x,y,z;

    If(n>0) {
        VERIFICA_ALGO(n-1);
        FOR(x=0; x< n; x++) {
            faz_algo;
        }
    }
}

```

```

c) VERIFICA_ALGO(int n) {
    INT x,y,z;

    If(n>1) {
        VERIFICA_ALGO(n/2);
        FOR(x=0; x< n; x++) {
            faz_algo;
        }
    }
}

```

4- Implemente os algoritmos de ordenação heapsort, quicksort e radixsort em programas que ordenem vetores de inteiros gerados aleatoriamente. Apresente uma análise de custos para cada algoritmo e em seguida considere que os algoritmos serão usados para (50%):

- a) ordenar vetores numéricos onde em média 90% dos elementos têm o mesmo valor e os demais apresentem valores distintos e distribuídos de maneira uniforme o longo do vetor.
- b) ordenar vetores onde não há elementos repetidos e chaves foram inseridas de maneira aleatória.

Apresente para cada item testes empíricos para avaliar o desempenho dos algoritmos em função do aumento do tamanho da entrada. Gere gráficos com as curvas de tempo de execução e avalie as curvas para tecer conclusões sobre o desempenho dos algoritmos

estudados. Ao final, com base na análise dos algoritmos e dos resultados dos testes, indique o melhor algoritmo para diferentes tamanhos de entrada e nas duas situações apresentadas.

O que entregar na questão: código impresso das funções implementadas; explicação dos algoritmos; análise de custos envolvidos de tempo e espaço; e relatório dos experimentos realizados com as conclusões obtidas e detalhes dos resultados que levaram a tais conclusões.