"A compreensão das questões faz parte da avaliação"

# Análise e Projeto de Algoritmos

Bacharelado em Ciência da Computação

1ª Avaliação Individual

6 de abril de 2015

**Nome:** ______________________________ **Período:**______

***Questão 1*: (1.0 ponto)**
Considerando todos os algoritmos de ordenação estudados em aula, complete as 5 posições vazias no quadro abaixo com as complexidades assintóticas corretas.

| Algoritmo | Pior caso | Melhor caso | Caso médio | Espaço |
|---|---|---|---|---|
| Insertion-Sort | $\theta(n^2)$ | | $\theta(n^2)$ | $\theta(n)$ |
| Merge-Sort | $\theta(n \lg n)$ | $\theta(n \lg n)$ | $\theta(n \lg n)$ | $\theta(n)$ |
| Selection-Sort | $\theta(n^2)$ | $\theta(n^2)$ | $\theta(n^2)$ | $\theta(n)$ |
| Heap-Sort | $O(n \lg n)$ | $O(n \lg n)$ | | $\theta(n)$ |
| Quick-Sort | | $\theta(n \lg n)$ | $\theta(n \lg n)$ | $\theta(n)$ |
| Counting-Sort | $\theta(k+n)$ | $\theta(k+n)$ | $\theta(k+n)$ | $\theta(k+n)$ |
| Radix-Sort | $\theta(d(n+k))$ | $\theta(d(n+k))$ | $\theta(d(n+k))$ | $\theta(d(n+k))$ |
| Bucket-Sort | | $\theta(n)$ | $\theta(n)$ | |

---

***Questão 2*: (1.5 pontos)**
Prove que $O(n^2) + O(n^2) + O(n^2) = O(3n^2)$.

---

***Questão 3*: (1.5 pontos)**
Aplique o método da árvore de recursão para determinar a solução da recorrência a seguir para uma constante $c > 0$.

$$T(n) = \begin{cases} \theta(1) & \text{, se } n \leq 3 \\ 3T(\lfloor \frac{n}{4} \rfloor) + \theta(n^2) & \text{, se } n \geq 4 \end{cases}$$

---

***Questão 4*: (2.0 pontos)**
Prove a corretude do algoritmo de Euclides, que determina o Máximo Divisor Comum (MDC) entre 2 números naturais. Lembre-se que $mdc(m, 0) = mdc(0, m) = m$ para todo $m \in \mathbf{N}$, com $m > 0$, e além disso, $mdc(x, y) = mdc(y, x \bmod y)$.

```
MDC(m,n) // supõe m != 0 ou n != 0
x <- m
y <- n
enquanto y != 0 faça
     r <- x mod y
     x <- y
     y <- r
devolva x
```

---

***Questão 5*: (2.0 pontos)**
Demonstre(prove), formalmente, que qualquer algoritmo de ordenação por comparação exige $\Omega(n \lg n)$ comparações no pior caso.

***Questão 6*:** (2.0 pontos)
Quando analisamos a versão aleatorizada do algoritmo Quick-Sort, obtida por meio da adoção de um particionamento aleatório, chegamos na recorrência apresentada a seguir. Mostre (prove) que $T(n) = O(n \lg n)$.

$$T(n) = \begin{cases} \theta(1) & , \text{se } n = 1 \\ \dfrac{1}{n}\left(\sum\limits_{k=0}^{n-1} \big(T(k) + T(n-1-k) + \theta(n)\big)\right) & , \text{se } n > 1 \end{cases}$$

**"Esta avaliação terá duração máxima de 3 horas"**