

Tópicos Avançados em Estrutura de Dados

Atividade 2

Bruna Galastri Guedes – 18.00189-0

Daniel Ughini Xavier – 18.00022-3

Leonardo Cury Haddad - 18.00442-3

Leonardo de Barros Rodrigues – 18.02401-7

Rodolfo Cochi Bezerra – 18.00202-0

Vítor Martin Simoni – 18.00050-9

15/03/2020

Questão 1

a) Tendo como base o seguinte algoritmo InsertionSort, implementado na linguagem C:

```
void insertionSortCrescente(int array[ ], int tamanho){
    int i, j, tmp;
    for (i = 1; i < tamanho; i++){
        j = i;
        while (j > 0 && array[j - 1] > array[j]){
            tmp = array[j];
            array[j] = array[j - 1];
            array[j - 1] = tmp;
            j--;
        }
    }
}
```

Foi realizada a análise de complexidade da seguinte forma:

O pior caso a ser considerado para o algoritmo acima seria pedir a ordenação de um vetor de tamanho "n" que estivesse em ordem decrescente, pois nesse caso seriam necessários $2(n-i)$ operações para cada elemento percorrido na sequência. Por exemplo, se um vetor com ordenação decrescente for percorrida pelo algoritmo, último elemento é trocado e comparado $(n-1)$ vezes, o penúltimo elemento, $(n-2)$ vezes, e assim por diante. Com isso, o número de operações realizadas é $2 \cdot (1 + 2 + \dots + n-2 + n-1)$, dependendo do número de elementos do vetor. Reescrevendo tal conta de forma genérica, temos:

$$\sum_{q=1}^p q = \frac{p(p+1)}{2}$$

E para o pior caso temos:

$$\frac{2(n-1)(n)}{2} = (n-1)(n) \rightarrow O(n^2)$$

Portanto o algoritmo possui complexidade (n^2) atuando no pior caso.