

Universidade Federal de Mato Grosso do Sul
Algoritmos e Programação II
Prof. Anderson Bessa da Costa

Atividade EaD (4 horas)
Generalização da Função de Ordenação por Flutuação com Ponteiros

Prezado/a Estudante!

Nesta atividade, abordaremos sobre o conceito de ponteiros. Um ponteiro é uma variável capaz de armazenar um endereço de memória ou o endereço de outra variável. Os conceitos de endereço e ponteiro são fundamentais em qualquer linguagem de programação. Em C/C++, esses conceitos se tornam ainda mais importantes pois são explícitos, por isso é essencial conhecê-los de forma adequada.

A utilização de ponteiros (às vezes chamados de apontadores) em linguagem C/C++ é uma das características que tornam a linguagem tão flexível e poderosa. Ao mesmo tempo é o seu calcanhar de aquiles, pois o seu emprego não é simples e muitas vezes acaba sendo propenso a erros.

Nessa atividade, você adaptará uma função de ordenação por flutuação fornecida, capaz de ordenar números inteiros, para que ordene elementos de quaisquer tipos. Essa adaptação exigirá a utilização correta dos conceitos de ponteiros, *casting* e ponteiro para função.

O passo a passo desta atividade consiste em:

- 1) Assista a videoaula <https://youtu.be/jmtJnsQVY3s>;
- 2) (Opcional) Leia o Apêndice D (pág. 147 a 151) do livro FEOFILOFF, Paulo. Algoritmos em Linguagem C. **Elsevier**, 2009 (disponível na biblioteca da UFMS);
- 3) Implemente em linguagem C/C++ a atividade descrita após este passo a passo;
- 4) Submeta a sua implementação no AVA Moodle da disciplina.

Esta atividade contabilizará 4 horas-aula de presença.

Bons estudos!

Seja o protótipo da função *qsort* em C:

```
void qsort(void *base, size_t nitems, size_t size, int (*compare)(const void *, const void*))
```

qsort é uma função da biblioteca padrão C (stdlib.h) que implementa um algoritmo de ordenação de vetor de objetos arbitrários de acordo com uma função de comparação fornecida pelo utilizador. O seu nome vem do algoritmo “*quicker sort*” (uma variante do quicksort devido a R. S. Scowen), que foi originalmente utilizado para o implementar na biblioteca C da Unix, embora a padrão C não exija que implemente o quicksort.

Para a função *qsort* ordenar qualquer tipo de dados, ela requer algumas informações: endereço base (*base*), quantidade de itens (*nitems*), tamanho do item (*size*) e uma função de comparação (**compare*).

O objetivo desta atividade é você generalizar o algoritmo de ordenação por flutuação (*bubble sort*), fornecido a seguir, tornando-o capaz de ordenar qualquer tipo de dado com as mesmas informações fornecidas à função *qsort*.

```
void bubble_sort(int v[], int n) {  
    int i, j, aux;  
  
    for (i = 0; i < n-1; i++) {  
        for (j = 0; j < n - i - 1; j++) {  
            if (v[j] > v[j+1]) {  
                aux = v[j];  
                v[j] = v[j+1];  
                v[j+1] = aux;  
            }  
        }  
    }  
}
```

A assinatura da função *bubble sort* deverá ser a mesma que a função *qsort*:

```
void bubble_sort(void *base, size_t nitems, size_t size, int  
(*compare)(const void *, const void*))
```