



Universidade Federal da Fronteira Sul - UFFS
Curso de Ciência da Computação

Cocktail Sort

Mauricio Catanio
Marcos Henrique Curtale Serafim

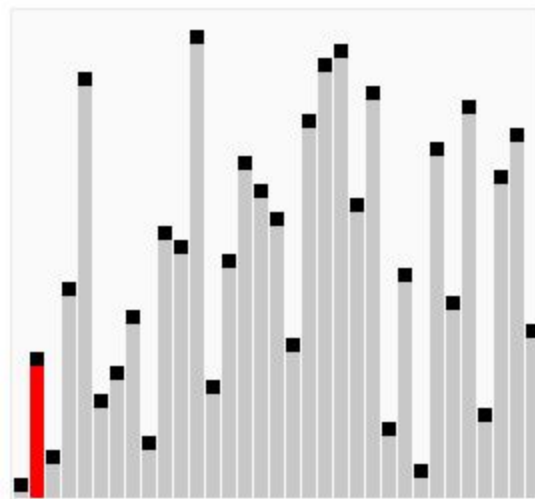
Cocktail Sort é um método de ordenação estável e baseado em comparações. É uma variação do conhecido bubble sort, onde a diferença reside na ordenação em ambas as direções que percorre (Esquerda para direita e direita para esquerda).

Na primeira percorrida pela lista de elementos (esquerda para a direita) o algoritmo efetua comparações a cada dois elementos de modo que o maior elemento ficará à direita. Já no retorno (esquerda para a direita) compara-se os elementos de modo ao menor elemento ficar a esquerda. Este procedimento é feito sequencialmente até que todos os elementos estejam ordenados.

Para controlar o movimento o algoritmo utiliza 5 variáveis de controle, denominadas **largura**, **base**, **topo**, **troca** e **i**. A variável largura é responsável por saber qual o tamanho total do vetor, enquanto base e topo são para determinação de onde até onde o vetor será percorrido.

A cada ida e volta o **topo** é decrementado em um e a **base** incrementada em um. Isto porque, como dito anteriormente, o algoritmo funciona de forma que o maior elemento sempre ficará a direita e o menor sempre à esquerda.

Quando neste percurso não ocorrem troca de posições, indica que o vetor está ordenado. A variável **troca** passa a ter valor zero e a função é finalizada.



Código em C

```
void cocktail_sort(int list[10]) {
    int length,bottom,top, swapped,i,aux;
    length=10;
    bottom = 0;
    top = length - 1;
    swapped = 0;
    while(swapped == 0 && bottom < top)//Se não houver troca de posições ou o
    ponteiro que
    {
        //sobe ultrapassar o que desce, o vetor esta ordenado
        swapped = 1;
        //Este for é a "ida" para a direita
        for(i = bottom; i < top; i = i++)
        {
            if(list[i] > list[i + 1]) //indo pra direita:testa se o próximo é maior
            { //indo pra direita:se o proximo é maior que o atual,
                //troca as posições
                aux=list[i];
                list[i]=list[i+1];
                list[i+1]=aux;
                swapped = 0;
            }
        }//fecha for
        // diminui o `top` porque o elemento com o maior valor
        // já está na direita (atual posição top)
        top = top - 1;
        //Este for é a "ida" para a esquerda
        for(i = top; i > bottom; i = i - 1)
        { if(list[i] < list[i - 1])
            {
                aux=list[i];
                list[i]=list[i-1];
                list[i-1]=aux;
                swapped = 0;
            }
        }
        //aumenta o `bottom` porque o menor valor já está
        //na posição inicial (bottom)
        bottom = bottom + 1;
    }
}
```

```
}//fecha while  
}// fim da função
```