

Método de ordenação por Intercalação: Mergesort – Aplicação em Vetores

Base do Método:

- 1 – divida o vetor em duas metades
- 2 – recursivamente ordene cada metade
- 3 – intercale os elementos das metades ordenadas

Exemplo: Seja o vetor inicial e as divisões sucessivas pela metade:

[68 29 33 30 6 34 26 23 65 59]									
[68 29 33 30 6]					[34 26 23 65 59]				
[68 29 33]			[30 6]		[34 26 23]			[65 59]	
[68 29]		[33]	[30]	[6]	[34 26]		[23]	[65]	[59]
[68]	[29]				[34] [26]				

Mistura das partes: [68] e [29] ==> [29 68]

[68 29 33 30 6 34 26 23 65 59]									
[68 29 33 30 6]					[34 26 23 65 59]				
[68 29 33]			[30 6]		[34 26 23]			[65 59]	
[29 68]		[33]	[30]	[6]	[34 26]		[23]	[65]	[59]
					[34] [26]				

Mistura das partes: [29 68] e [33] ==> [29 33 68]

[68 29 33 30 6 34 26 23 65 59]									
[68 29 33 30 6]					[34 26 23 65 59]				
[29 33 68]			[30 6]		[34 26 23]			[65 59]	
			[30]	[6]	[34 26]		[23]	[65]	[59]
					[34] [26]				

Mistura das partes: [30] e [6] ==> [6 30]

[68 29 33 30 6 34 26 23 65 59]									
[68 29 33 30 6]					[34 26 23 65 59]				
[29 33 68]			[6 30]		[34 26 23]			[65 59]	
					[34 26]		[23]	[65]	[59]
					[34] [26]				

Mistura das partes: [29 33 68] e [6 30] =====> [6 29 60 33 68]

[68 29 33 30 6 34 26 23 65 59]
[6 29 60 33 68] [34 26 23 65 59]
[34 26 23] [65 59]
[34 26] [23] [65] [59]
[34] [26]

Mistura das partes: [34] e [26] =====> [26 34]

[68 29 33 30 6 34 26 23 65 59]
[6 29 60 33 68] [34 26 23 65 59]
[34 26 23] [65 59]
[26 34] [23] [65] [59]

Mistura das partes: [26 34] e [23] =====> [23 26 34]

[68 29 33 30 6 34 26 23 65 59]
[6 29 60 33 68] [34 26 23 65 59]
[23 26 34] [65 59]
[65] [59]

Mistura das partes: [65] e [59] =====> [59 65]

[68 29 33 30 6 34 26 23 65 59]
[6 29 60 33 68] [34 26 23 65 59]
[23 26 34] [59 65]

Mistura das partes: [23 26 34] e [59 65] =====> [23 26 34 59 65]

[68 29 33 30 6 34 26 23 65 59]
[6 29 60 33 68] [23 26 34 59 65]

Mistura das partes: [6 29 60 33 68] e [23 26 34 59 65] =====>

[6 23 26 29 30 33 34 59 65 68]

Codificação em Linguagem C

```
void intercala (int v[], int e, int m, int d){

    int *r; //vetor auxiliar para armazenar a mistura
    int i,j,k;

    r = (int *) malloc (((d+1)-e)*sizeof(int));

    i=e;
    j=m+1;
    k=0;

    while ((i<=m)&&(j<=d)){          // intercala enquanto nenhuma das partes do vetor
        if(v[i]<=v[j]){              // foi consumida totalmente
            r[k]=v[i];
            i++;
        }
        else{
            r[k]=v[j];
            j++;
        }
        k++;
    }

    while (i<=m) {r[k]=v[i];i++;k++;} //termina de intercalar se sobrou à esquerda
    while (j<=d) {r[k]=v[j];j++;k++;} //termina de intercalar se sobrou à direita

    //Cópia do vetor auxiliar (contendo os dados ordenados) sobre o vetor original
    j=0;
    for(i=e;i<=d;i++){ v[i]=r[j]; j++; }

    free(r);
}

void mergesort (int v[], int e, int d){
    int m;

    if (e<d){
        m = (e+d)/2;
        mergesort(v,e,m);
        mergesort(v,m+1,d);
        intercala(v,e,m,d);
    }
}
```

=====