Júlia Dambrós

Teste de mesa pág. 17

N	[0]	[1]	[2]	[3]	[4]	I	J	aux	j>=0	Aux<v[j]	V[j+1]=v[j]	j--	V[j+1]=aux
5	8	5	7	1	4	1	0	1 5	V	V	5=8 8	-1	5
							-1		F	F			
	5	8	7	1	4	2	1	2 7	V	V	2=1 7=8 8	0	
							0		V	F			7
	5	7	8	1	4	3	2	3 1	V	V	3=2 1=8 8	1	
							1		V	V	3=1 1=7 7	0	
							0		V	V	3=0 1=5 5	-1	1
	1	5	7	8	4	4	3	4 4	V	V	4=3 4=8 8	2	
							2		V	V	4=2 4=7 7	1	
							1		V	V	4=1 4=5 5	0	
							0		V	F		-1	4
	1	4	5	7	8								

Atividade Proposta (insertion)

I.

N	[0]	[1]	[2]	[3]	[4]	I	J	aux	j>=0	Aux<v[j]	V[j+1]=v[j]	j--	V[j+1]=aux
	7	2	5	4	9	1	0	1	V	V	[1]=7	-1	[1]<=7
	2	7	5	4	9	2	1	2	V	V	[2]=7	0	

							0		V	F	[1]=5	-1	[1]<=5
	2	5	7	4	9	3	2	3	V	V	[3]=7	1	
							1		V	V	[2]=5	0	
							0		V	F	[1]=4	-1	[1]<=4
	2	4	5	7	9	4	3	4	V	F		2	
							2		V	F		1	
							1		V	F		0	
							0		V	F		-1	
	2	4	5	7	9								

N	[0]	[1]	[2]	[3]	[4]	I	J	aux	j>=0	Aux<v[j]	V[j+1]=v[j]	j--	V[j+1]=aux
	7	5	5	3	9	1	0	1	V	V	[1]=7	-1	[1]=7
	5	7	5	3	9	2	1	2	V	V	[2]=7	0	
							0		V	F			[1]=5
	5	5	7	3	9	3	2	3	V	V	[3]=7	1	
							1		V	V	[2]=5	0	
							0		V	V	[1]=5	-1	[1]=3
	3	5	5	7	9	4	3	4	V	F		2	
							2		V	F		1	
							1		V	F		0	
							0		V	F		-1	
	3	5	5	7	9								

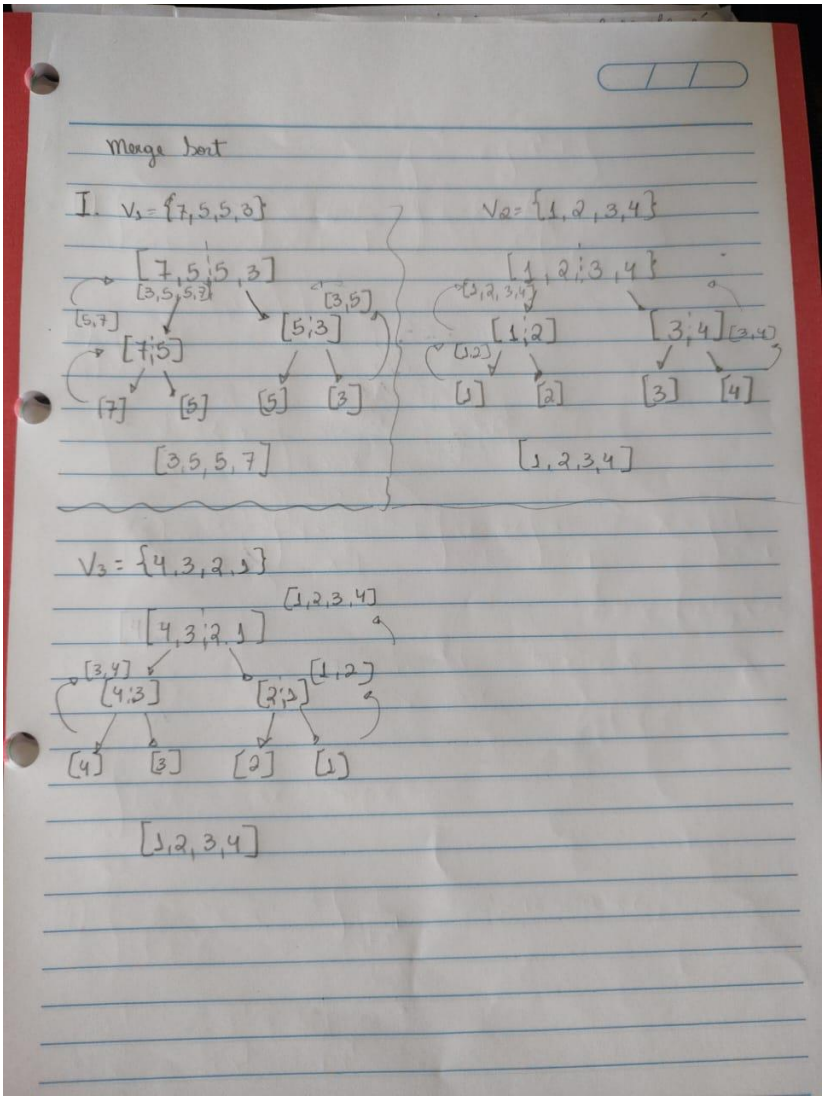
N	[0]	[1]	[2]	[3]	I	J	aux	j>=0	Aux<v[j]	V[j+1]=v[j]	j--	V[j+1]=aux
	1	2	3	4	1	0	1	V	F			0=0
					2	1	2	V	F			1=1
					3	2	3	V	F			2=2
					4	3	4	V	F			3=3
	1	2	3	4								

N	[0]	[1]	[2]	[3]	I	J	aux	j>=0	Aux<v[j]	V[j+1]=v[j]	j--	V[j+1]=aux
	4	3	2	1	1	0	1	V	V	[1]=4	-1	[1]=4
	3	4	2	1	2	1	2	V	V	[2]=4	0	
						0		V	V	[1]=3	-1	
	2	3	4	1	3	2	3	V	V	[3]=4	1	
						1		V	V	[2]=3	0	
						0		V	V	[1]=2	-1	
	1	2	3	4								

II. 10 comparações e 5 trocas

Atividade Proposta (merge)

I.



II.

Número de chamadas recursivas: 7

Número de trocas: 0 (nenhuma troca, pois v2 já está ordenado)

III.

```
void merges(int *v, int inicio, int fim) {
   if (inicio < (fim - 1)) {
      int meio = (inicio + fim) / 2;
      merges(v, inicio, meio);
      merges(v, meio, fim);
      intercala(v, inicio, meio, fim);
   }
}

void intercala(int *v, int inicio, int meio, int fim) {
   int *w, i = inicio, j = meio, k = 0;
   w = new (nothrow) int[fim];

   while (i < meio && j < fim) {
      // Modificação para ordenação decrescente
      if (v[i] > v[j]) {
         w[k] = v[i];
         i++;
      } else {
         w[k] = v[j];
         j++;
      }
      k++;
   }

   while (i < meio) {
      w[k] = v[i];
```

```
        i++;
        k++;
    }

    while (j < fim) {
        w[k] = v[j];
        j++;
        k++;
    }

    for (k = 0; k < fim - inicio; k++)
        v[inicio + k] = w[k];

    delete[] w; // Correção para usar delete[]
    w = nullptr; // Correção para atribuir nullptr
}
```

1) {2, 3, 4, 5, 6, 6, 7, 8, 9}