

I.

N	[0]	[1]	[2]	[3]	[4]	l	Menor	J	V[j]<[menor]	V[menor] ! v[i]
5	7	2	5	4	9	0	01	1	V	
							1	2	F	
							1	3	F	
							1	4	F	
										V 27
	2	7	5	4	9	1	1	2	V	
							2	3	F	
							2	4	F	
							2			V 257
	2	5	7	4	9	2	2	3	V	
							3	4	F	
										V 2537
	2	5	4	7	9	3	3	4	F	
							3			F
							4	4		F
										25479 (teria que rodar mais uma vez mas isso seria na chamada da função)

N	[0]	[1]	[2]	[3]	[4]	l	Menor	J	V[j]<[menor]	V[menor] ! v[i]
5	7	5	5	3	9	0	0	1	V	
							1	2	F	
							1	3	V	
							3	4	F	
	3	5	5	7	9	1	1	2	F	
							1	3	F	

							1	4	F	
	3	5	5	7	9	2	2	3	F	
							2	4		
	3	5	5	7	9	3	3	4	F	
										35579

N	[0]	[1]	[2]	[3]	I	Menor	J	V[j]<[menor]	V[menor] ! v[i]
4	1	2	3	4	0	0	1	F	
						0	2	F	
						0	3	F	
					1	1	2	F	
						1	3	F	
					2	2	3	F	
									1234

N	[0]	[1]	[2]	[3]	I	Menor	J	V[j]<[menor]	V[menor] ! v[i]
4	4	3	2	1	0	0	1	V	
						1	2	V	
						2	3	V	
						1			1324
	1	3	2	4	1	1	2	V	
						2	3	F	
									1234
	1	2	3	4	2	2	3	F	
									1234

II. Vai dar errado pois ele vai comparar com uma casa inexistente.

III. Ele vai considerar como falso (não será menor) e continuará normal.

IV. V3 não terá troca e em V4 duas trocas

V.

```

void selection(int *v, int n) {

int i, j, maior

for (i = 0; i < n - 1; i++) {

maior = i;

for (j = i + 1; j < n; j++) {

if (v[j] > v[maior]) {

maior = j;

}

}

if (v[maior] != v[i]) {

troca(&v[maior], &v[i]);

}

}

}

```

Pág. 16

I.

N	[0]	[1]	[2]	[3]	[4]	I (N-1)	J (=0)	J<i	V[J]>V[J+1]	J++
5	7	2	5	4	9	4	0	V	V	1
	2	7					1	V	V	2
	2	5	7				2	V	V	3
	2	5	4	7			3	V	F	
	2	5	4	7	9	3	0	V	F	1
							1	V	V	2
	2	4	5				2	V	F	3
							3	F	F	
	2	4	5	7	9	2	0	V	F	1
							1	V	F	2
							2	F		
						1	0	V	F	1
							1	F	F	

						0	0	F	F	
	2	4	5	7	9					

N	[0]	[1]	[2]	[3]	[4]	I (N-1)	J (=0)	J<i	V[J]>V[J+1]	J++
5	7	5	5	3	9	4	0	V	V	1
	5	7					1	V	V	2
		5	7				2	V	V	3
			3	7			3	V	F	
	5	5	3	7	9	3	0	V	F	1
							1	V	V	2
		3	5				2	V	F	3
	5	3	5	7	9	2	0	V	V	1
	3	5					1	V	F	
	3	5	5	7	9	1	0	V	F	
	3	5	5	7	9					

N	[0]	[1]	[2]	[3]	I (N-1)	J (=0)	J<i	V[J]>V[J+1]	J++
4	1	2	3	4	3	0	V	F	1
						1	V	F	2
						2	V	F	3
					2	0	V	F	1
						1	V	F	2
					1	0	V	F	1
					0	0	F	F	
	1	2	3	4					

N	[0]	[1]	[2]	[3]	I (N-1)	J (=0)	J<i	V[J]>V[J+1]	J++
4	4	3	2	1	3	0	V	V	1
	3	4				1	V	V	2
		2	4			2	V	V	3

	3	2	1	4	2	0	V	V	1
	2	3				1	V	V	2
	2	1	3	4	1	0	V	V	1
	1	2	3	4	0	0	F	F	
	1	2	3	4					

II. 10 comparações e 5 trocas

III. pegando como exemplo o v2: no selection 10 comparações e uma troca e no bubble 10 comparações e 5 trocas.

IV.

```
#include <iostream>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
void troca(int *a, int *b) {
```

```
    int temp = *a;
```

```
    *a = *b;
```

```
    *b = temp;
```

```
}
```

```
void bubbleRecurso(int *v, int n) {
```

```
    if (n == 1) {
```

```
        return;    }
```

```
    for (int i = 0; i < n - 1; i++) {
```

```
        if (v[i] > v[i + 1]) {
```

```
            troca(&v[i], &v[i + 1]);
```

```
        }
```

```
    }
```

```
    bubbleRecurso(v, n - 1);
```

```
}
```

```
int main() {
```

```
    int arr[] = {9, 8, 7, 6, 5, 4, 3, 2, 1};
```

```
int n = sizeof(arr) / sizeof(arr[0]);  
bubbleRecurso(arr, n);  
cout<<"Array ordenado: "<<endl;  
for (int i = 0; i < n; i++) {  
    cout<< arr[i]<<endl;  
}  
return 0;  
}
```