

Ordenação

Ordenação por seleção

Para esse tipo de ordenação, o menor do array deve ser o primeiro elemento. É necessário procurar o menor e colocá-lo no primeiro índice, e depois repetir essa operação com o resto do array. É uma ordenação por força bruta. Possui complexidade $O(n^2)$.

```
#include <bits/stdc++.h>

using namespace std;

int main(){
    int n;
    cin >> n;
    int a[n];
    for (int i = 0; i < n; i++) cin >> a[i];
    for (int i = 0; i < n; i++){
        int aux = i;
        int menor = a[i];
        for (int j = (i+1); j < n; j++){
            if (a[j] < menor){
                menor = a[j];
                aux = j;
            }
        }
        int temp = a[i];
        a[i] = menor;
        a[aux] = temp;
    }
    for (int i = 0; i < n-1; i++) cout << a[i] << " ";
    cout << a[n-1] << endl;
}
```

Ordenação por inserção

Separa o array em 2: não ordenado e ordenado. Remove o elemento do array não ordenado e insere no ordenado, verificando os índices corretos para se colocar o novo elemento.

num=	3	2	8	1	5	4	6	7
i=0	3	2	8	1	5	4	6	7
i=1	3	2	8	1	5	4	6	7
i=2	2	3	8	1	5	4	6	7
i=3	2	3	8	1	5	4	6	7
i=4	1	2	3	8	5	4	6	7
i=5	1	2	3	5	8	4	6	7
i=6	1	2	3	4	5	8	6	7
i=7	1	2	3	4	5	6	8	7
	1	2	3	4	5	6	7	8

```
#include <bits/stdc++.h>

using namespace std;

int main(){
    int n;
    cin >> n;
    int a[n];
    int a_copy[n];
    for (int i = 0; i < n; i++) cin >> a[i];
    for (int i = 0; i < n; i++){
        a_copy[i] = a[i];
        for (int j = 0; j < i; j++){
            if (a_copy[i] < a_copy[j]){
                int aux = a_copy[j];
                a_copy[j] = a_copy[i];
                a_copy[i] = aux;
            }
        }
    }
}
```

```
    }  
}  
for (int i = 0; i < n; i++) cout << a_copy[i] << " ";  
    cout << endl;  
}
```