Ordenação

Ordenação por seleção

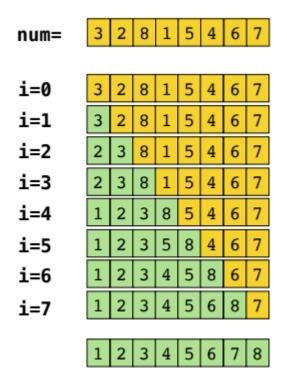
Para esse tipo de ordenação, o menor do array deve ser o primeiro elemento. É necessário procurar o menor e colocá-lo no primeiro índice, e depois repetir essa operação com o resto do array. É uma ordenação por força bruta. Possui complexidade O(n^2).

```
#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

Ordenação 1

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.



```
#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;
            }
```

Ordenação 2

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

Ordenação 3