Curso de Introdução à Programação de Computadores

Dia 4

Tarefa 1 - Imprime Vetor Invertido

- Escreva um programa que:
  - Leia uma linha contendo um inteiro N, (máximo 100).
  - A seguir leia uma linha com N inteiros (separados por espaço).
  - Imprimir uma linha contendo a sequência de inteiros na ordem inversa em que foram lidos. Após cada número da sequência, imprima um espaço em branco.

Tarefa 2 - Inverte Vetor Antes de Imprimir

- Escreva um programa que:
  - Leia uma linha contendo um inteiro N, (máximo 100).
  - A seguir leia uma linha com N inteiros (separados por espaço) e armazene em um vetor.
  - Inverta a ordem dos elementos do vetor.
  - Imprimir uma linha, contendo os elementos do vetor invertido. Após cada número da sequência, imprima um espaço em branco.

Inversão de vetor em C++

- Não precisamos implementar a inversão toda vez que precisamos usá-la!

- Comando reverse é utilizado para inverter os elementos de um vetor

- Para inverter um vetor vet com n elementos:

```
reverse(vet, vet + n);
```

Inversão de vetor em C++

```cpp
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int vet[100], i, n;

    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &vet[i]);
    }

    reverse(vet, vet + n);

    for (i = 0; i < n; i++) {
        printf("%d ", vet[i]);
    }

    printf("\n");

    return 0;
}
```

Tarefa 3 - Verifica Se Sequência É Ordenada

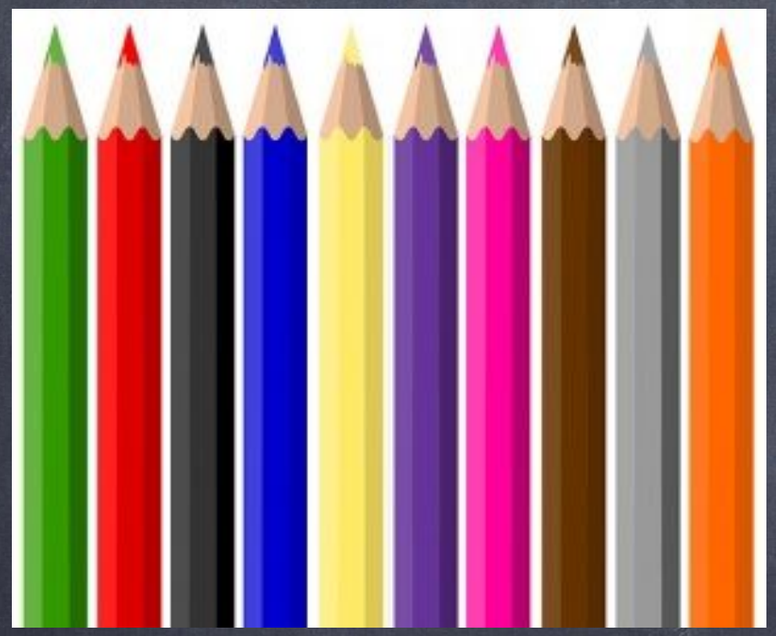
- Escreva um programa que:
  - Leia uma linha contendo um inteiro N, (2 ≤ N ≤ 100).
  - A seguir leia uma linha com N inteiros (separados por espaço); todos os valores são distintos, maiores ou iguais a zero.
  - Imprima uma linha contendo 'S' se a sequência lida está ordenada crescentemente, ou 'N' caso contrário.

Tarefa 4 - Ordena Sequência

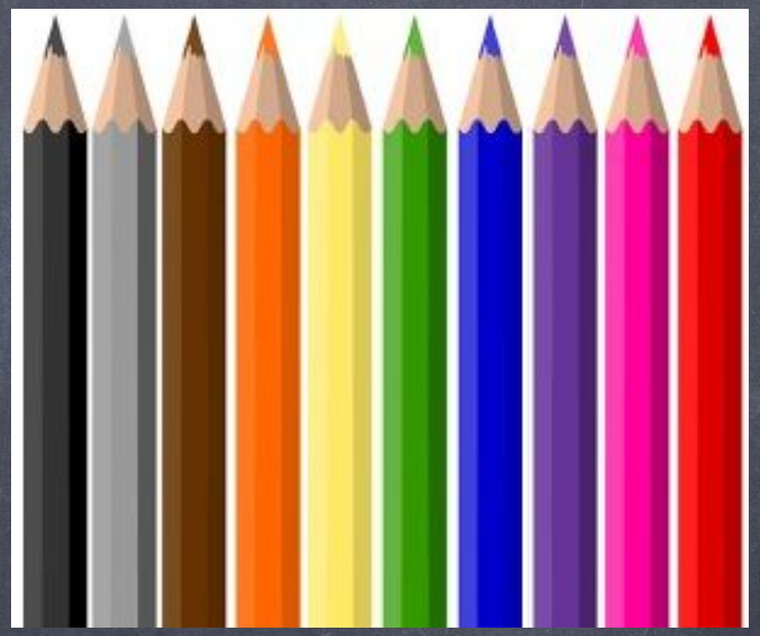
- Escreva um programa que:
  - Leia uma linha contendo um inteiro N, (2 ≤ N ≤ 100).
  - A seguir leia uma linha com N inteiros (separados por espaço); todos os valores são distintos, maiores ou iguais a zero.
  - Imprima uma linha contendo a sequência lida, ordenada crescentemente. Após cada número da sequência, imprima um espaço em branco.

Ordenação por seleção

- Queremos ordenar estes lápis:

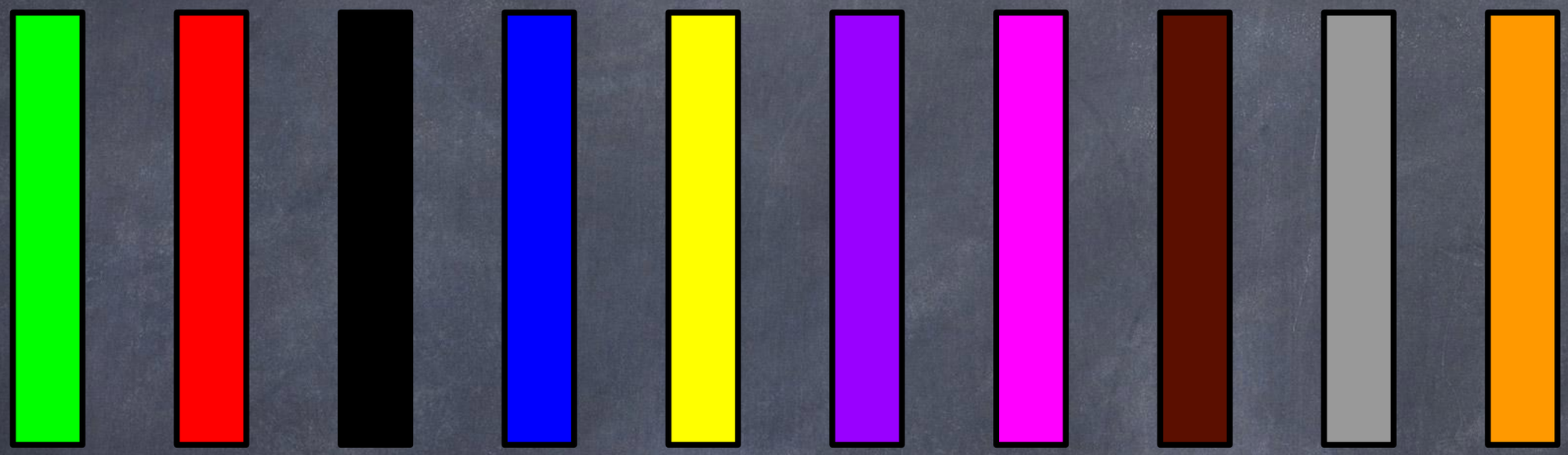


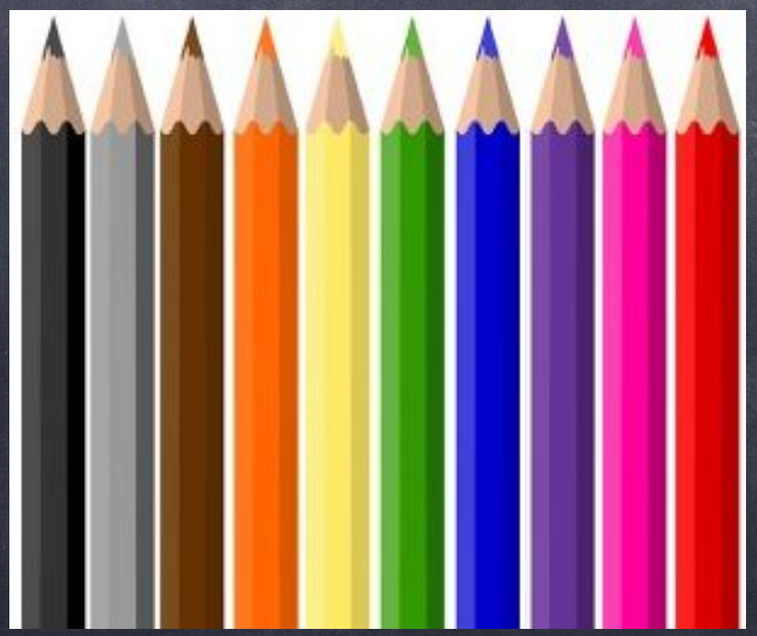
- Nesta ordem:



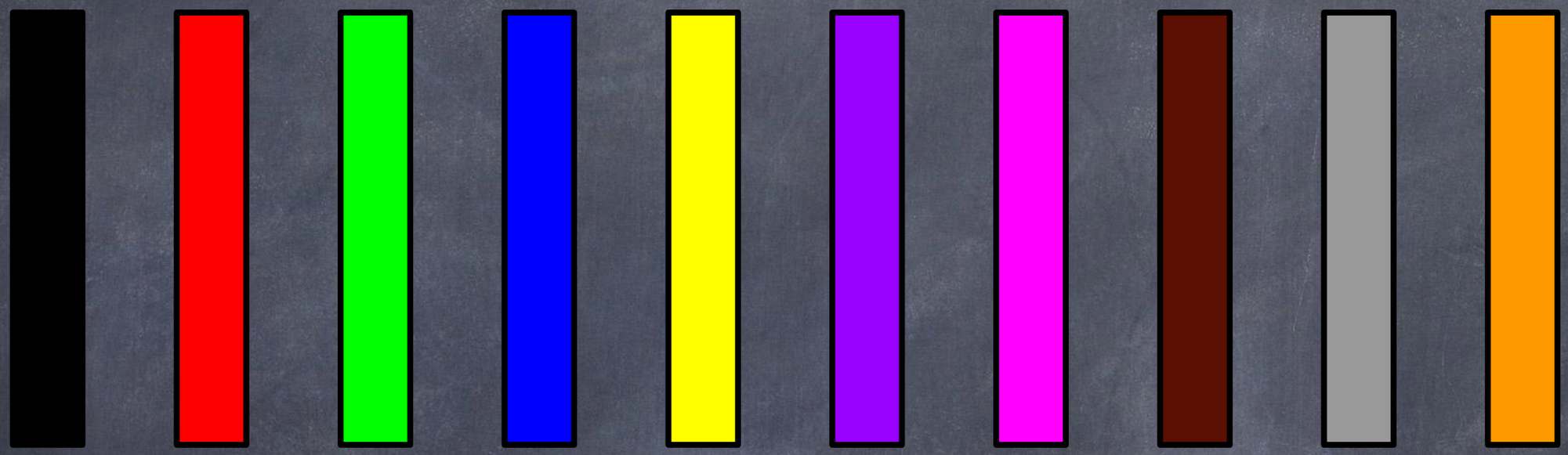
- Mas para colocar um lápis em uma nova posição da caixa, temos que trocar ele de lugar com o lápis que está agora naquela posição

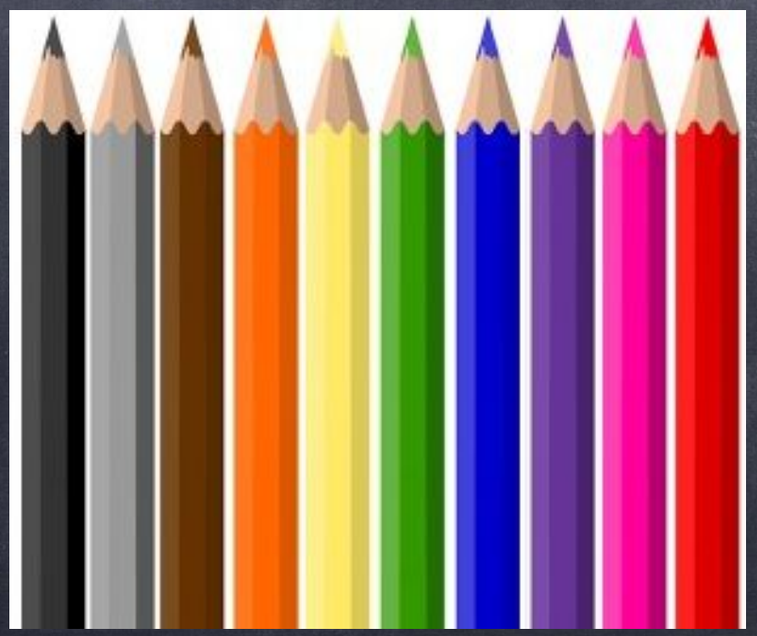
Ordenação por seleção



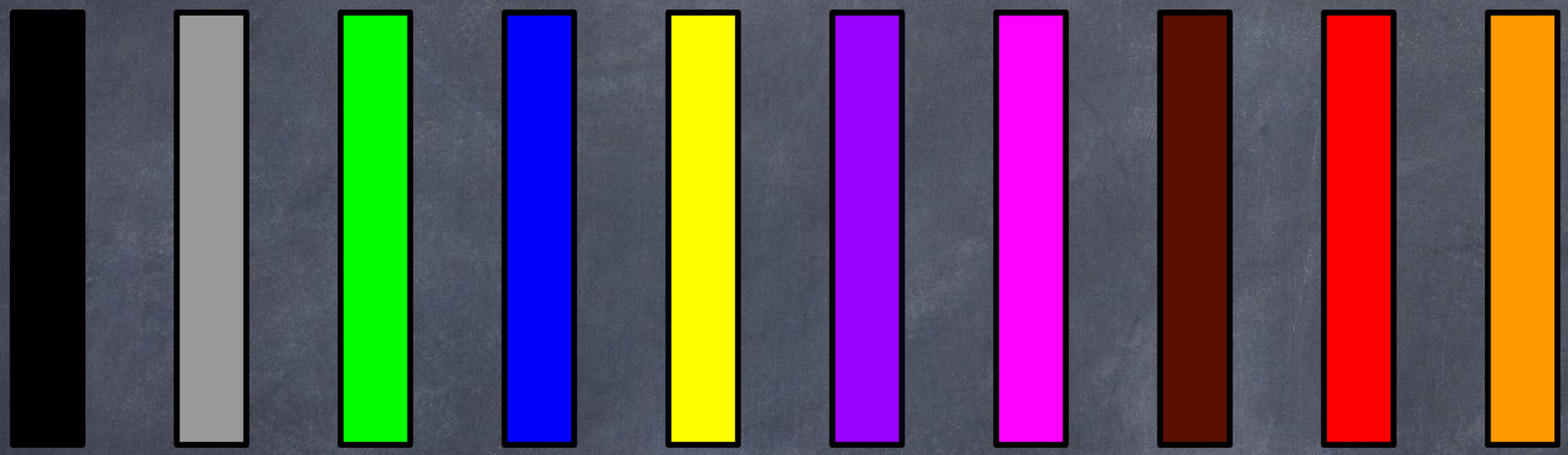


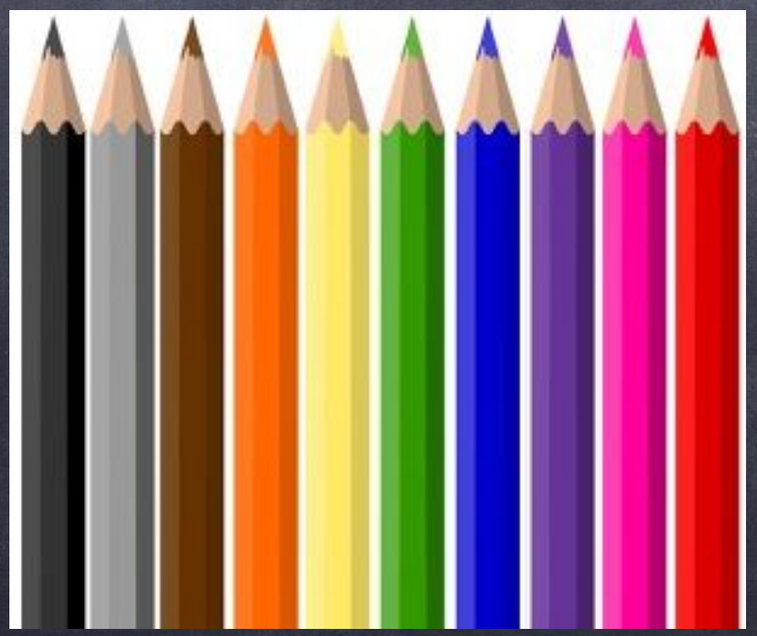
Ordenação por seleção



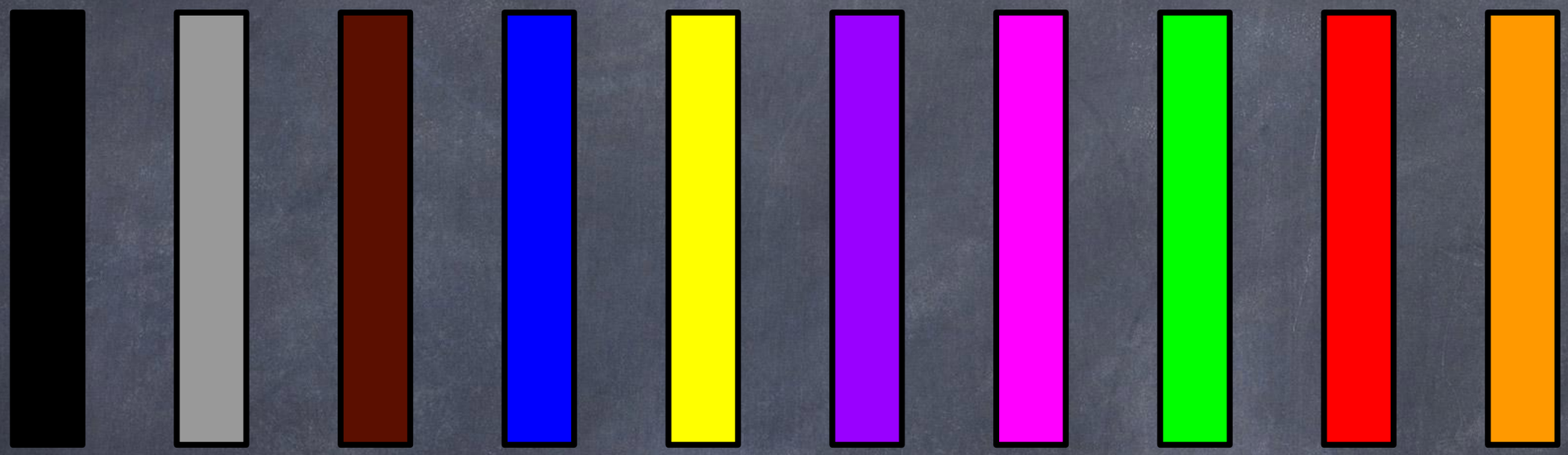


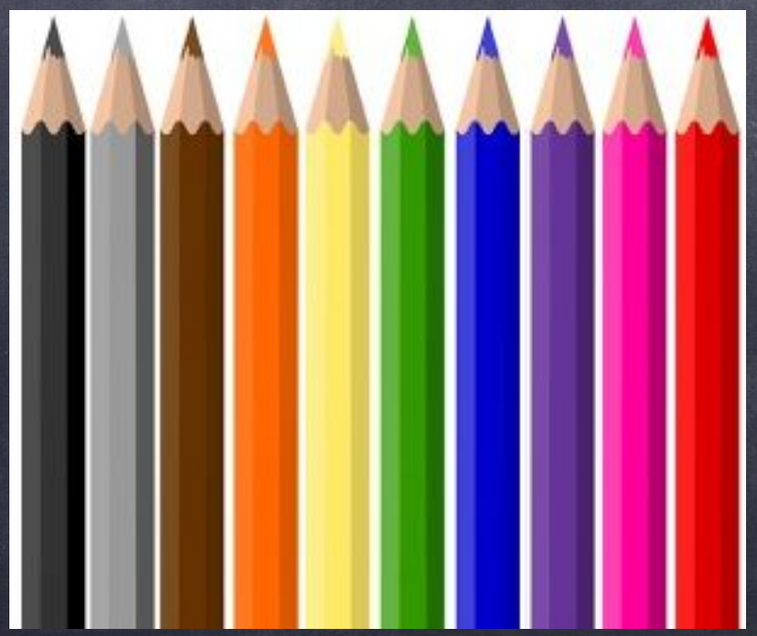
Ordenação por seleção



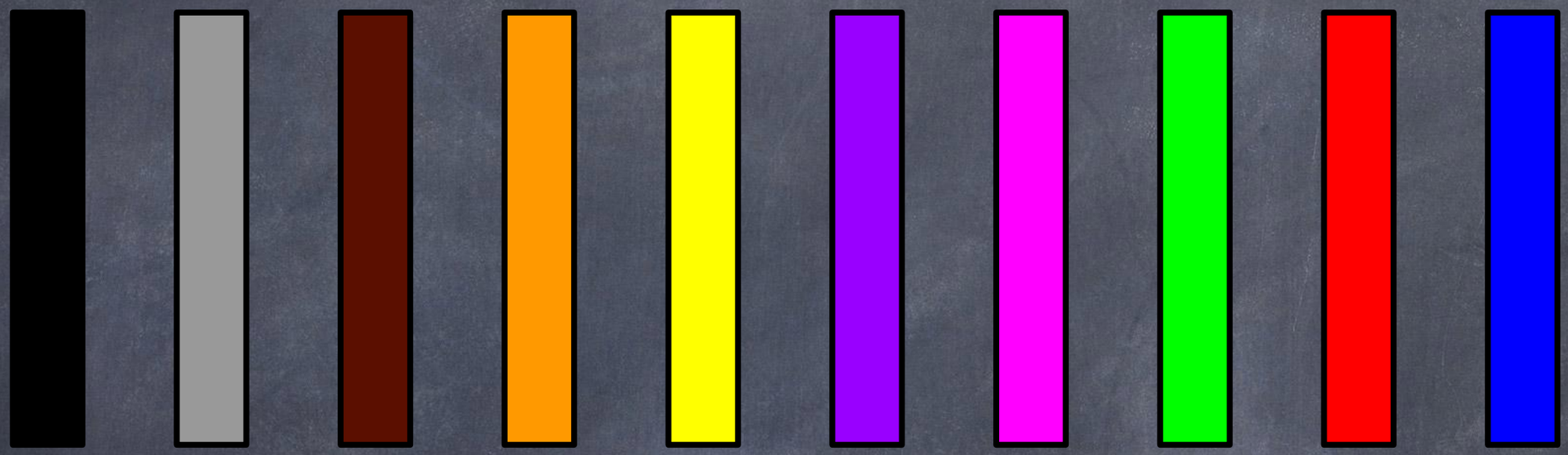


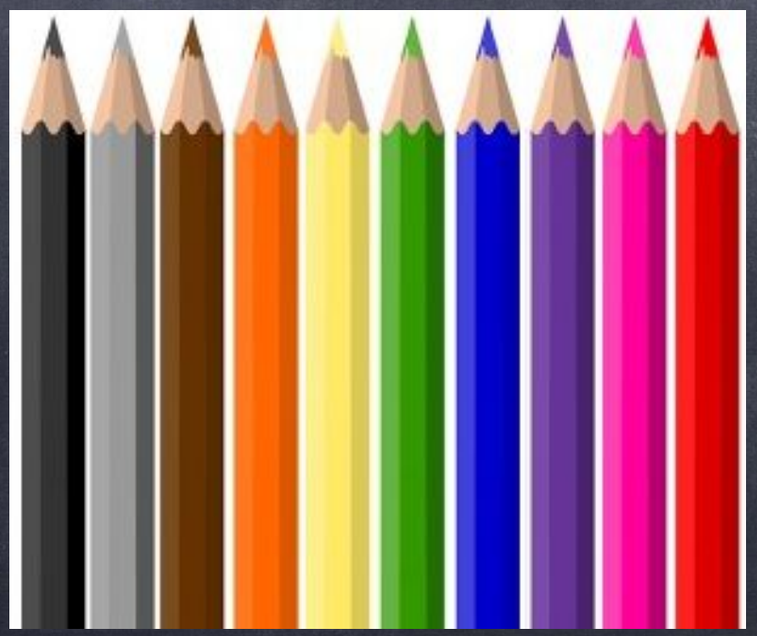
Ordenação por seleção



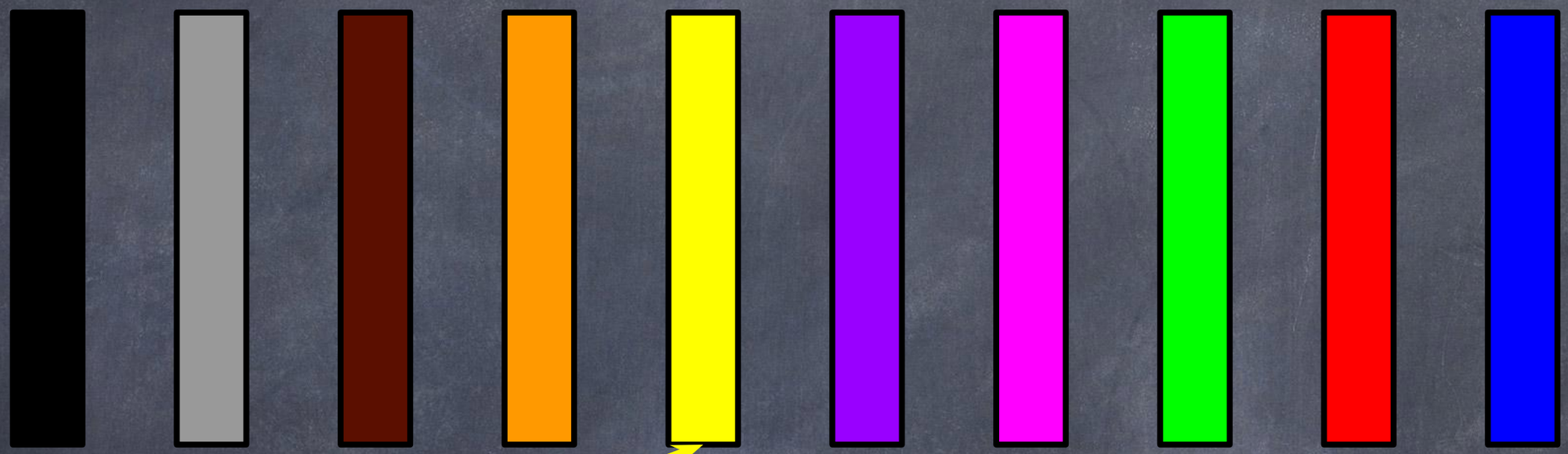


Ordenação por seleção

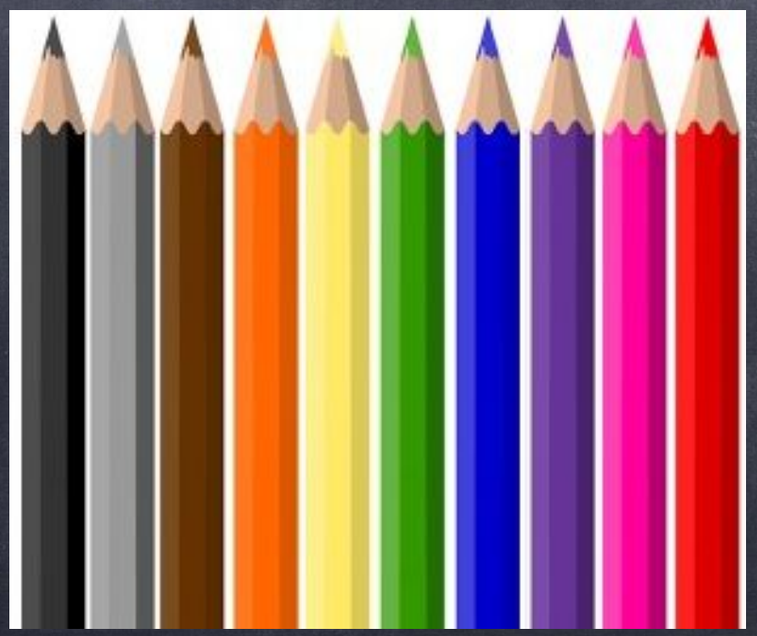




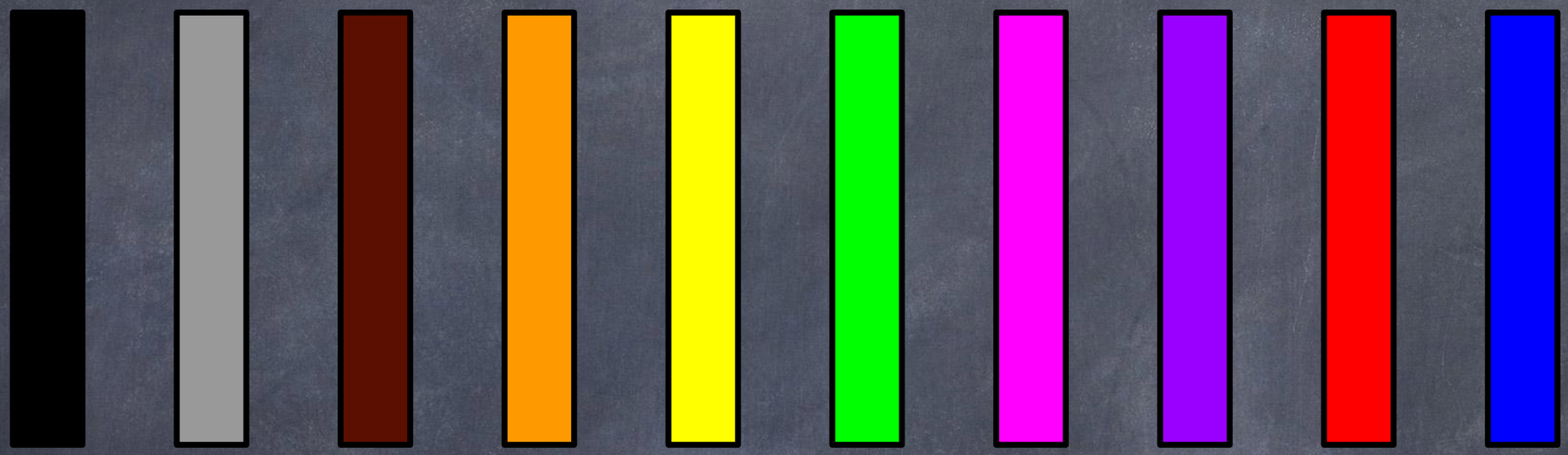
Ordenação por seleção

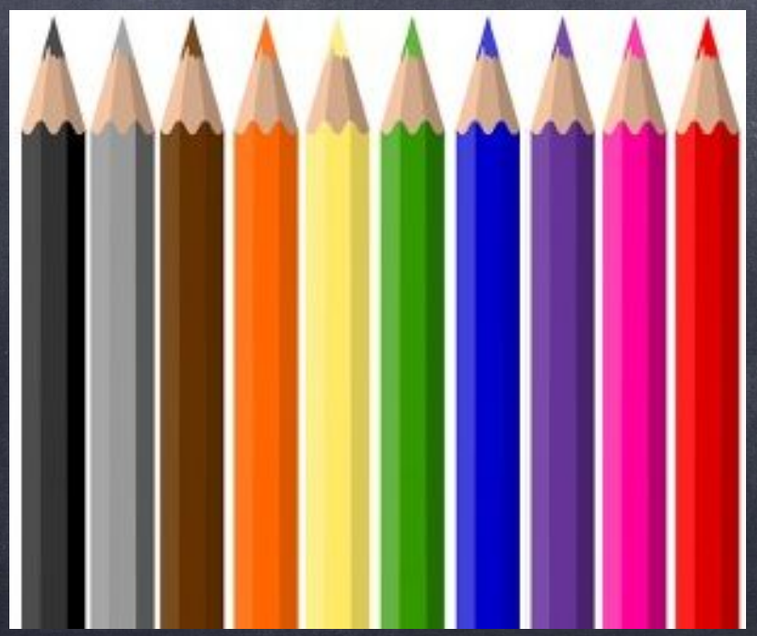


Já está na posição certa

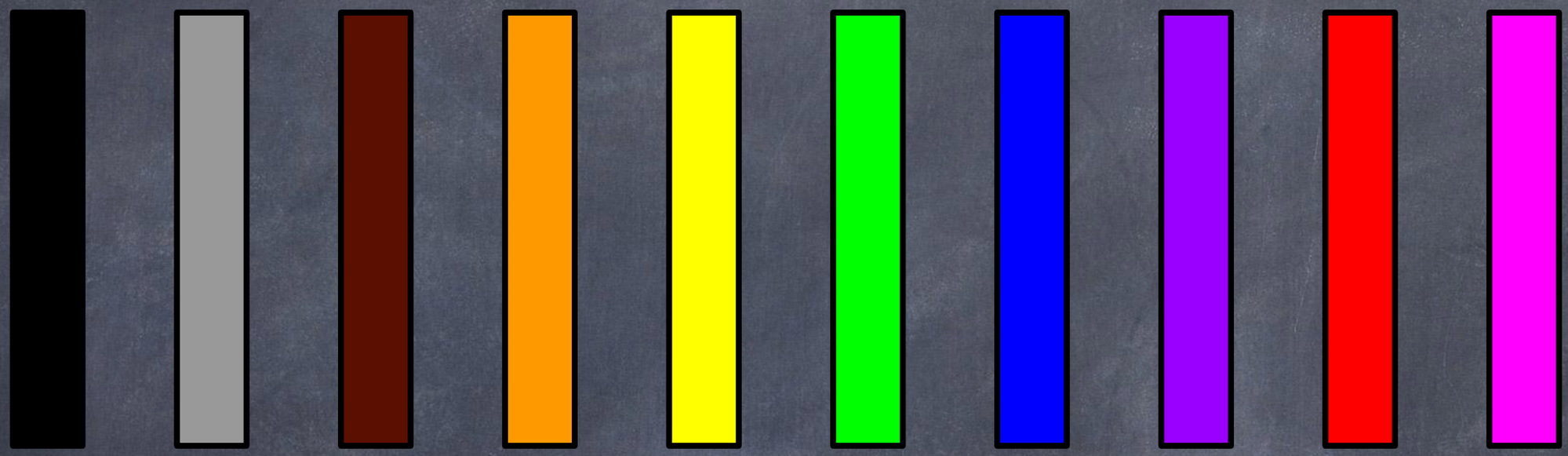


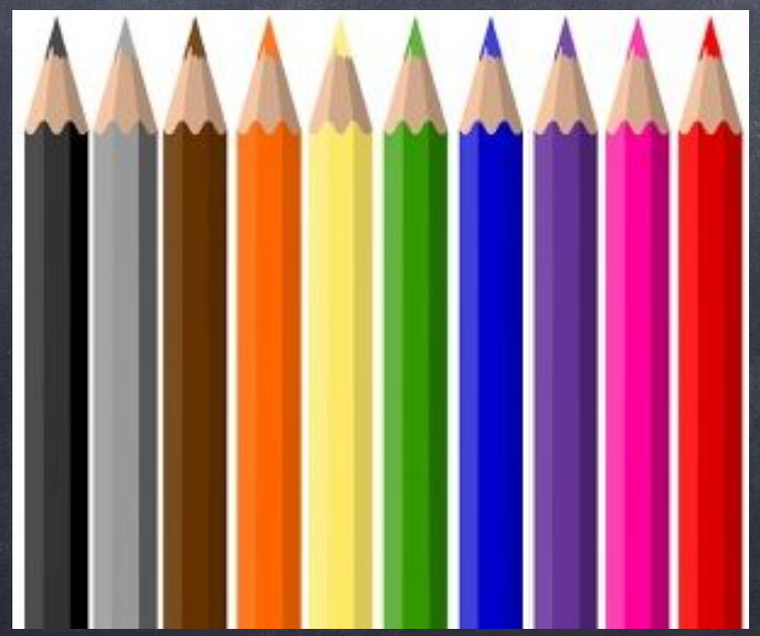
Ordenação por seleção



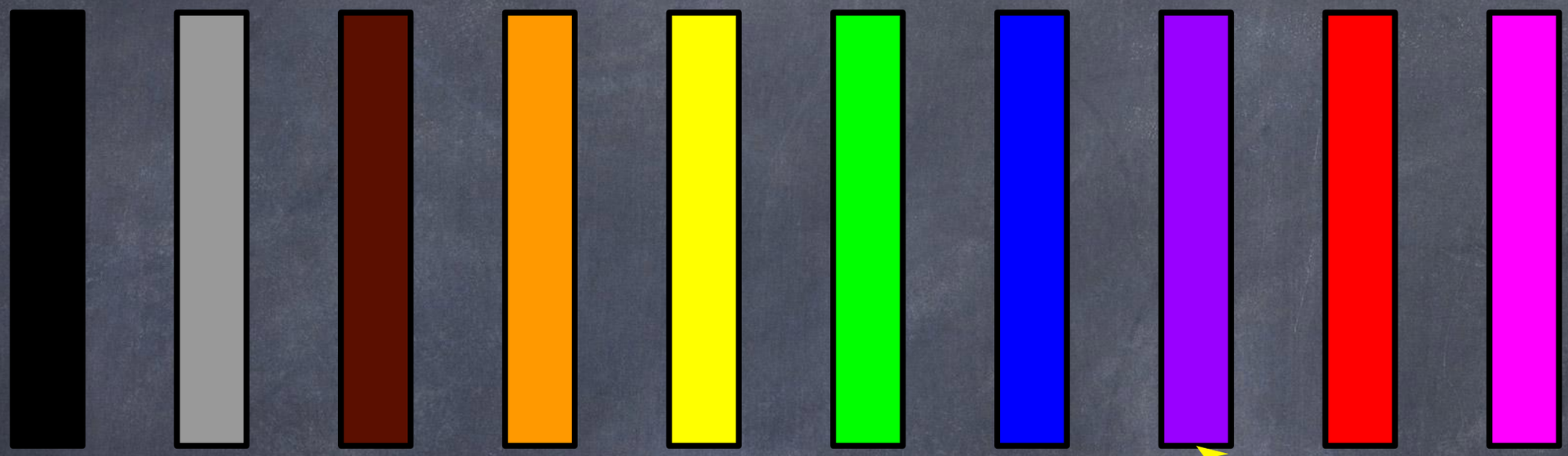


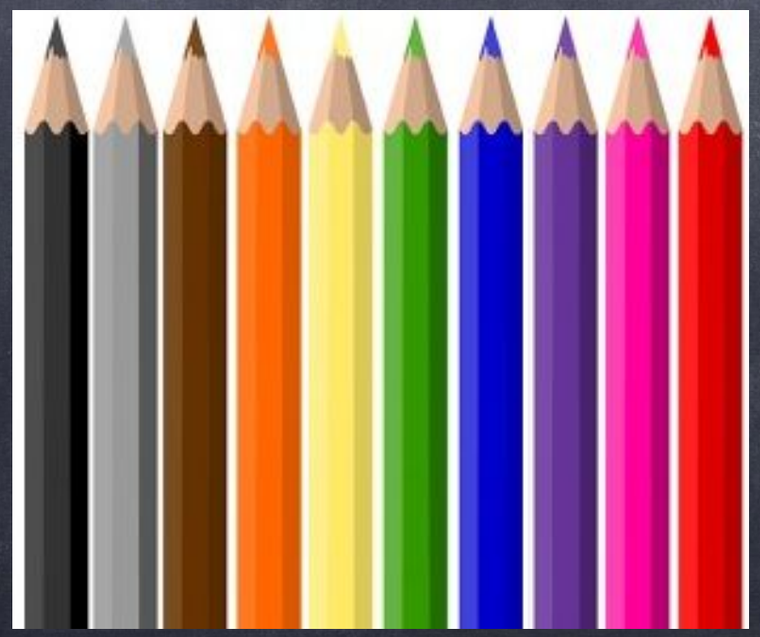
Ordenação por seleção





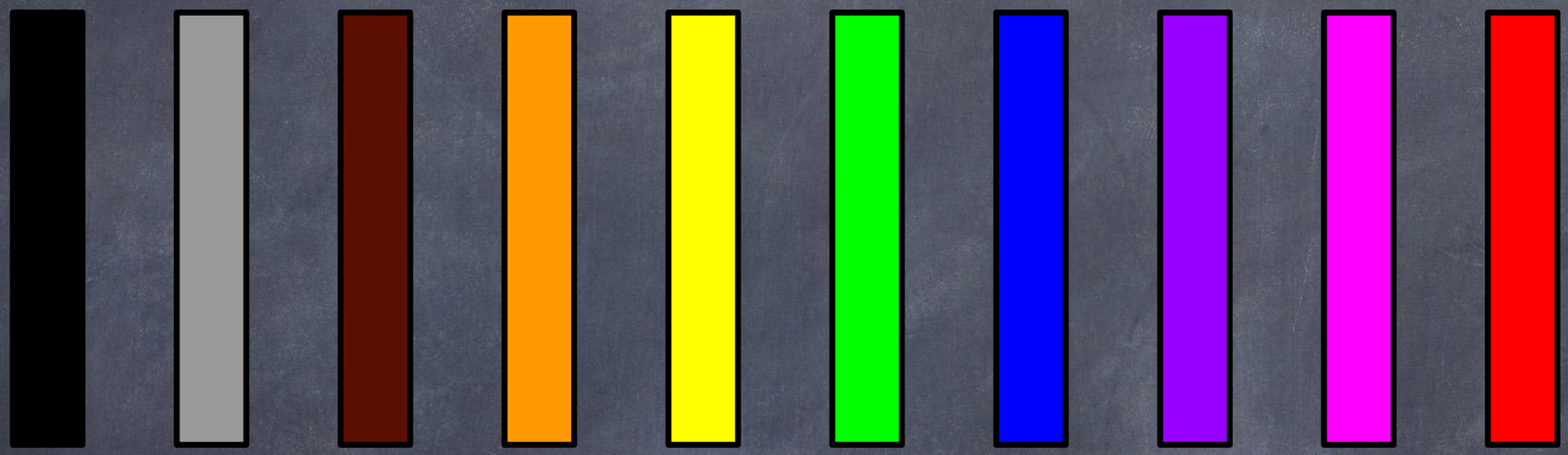
Ordenação por seleção

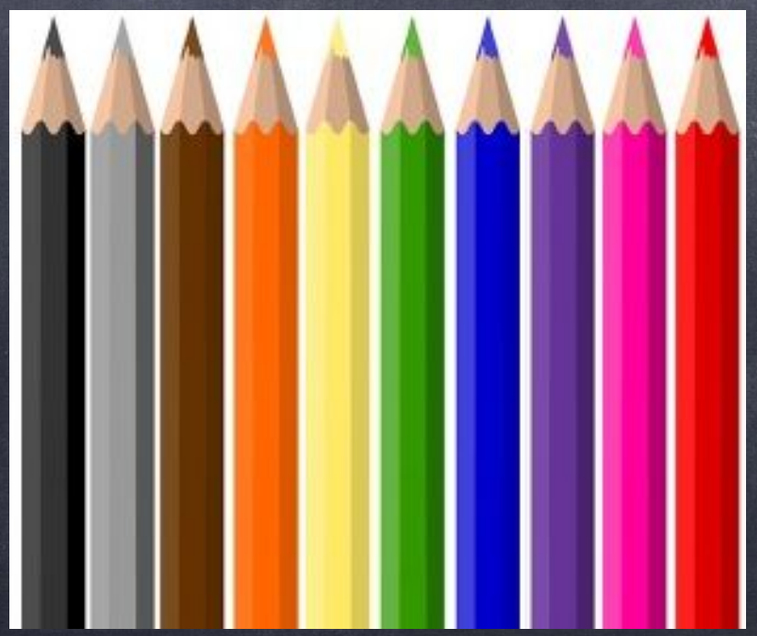




Já está na posição certa

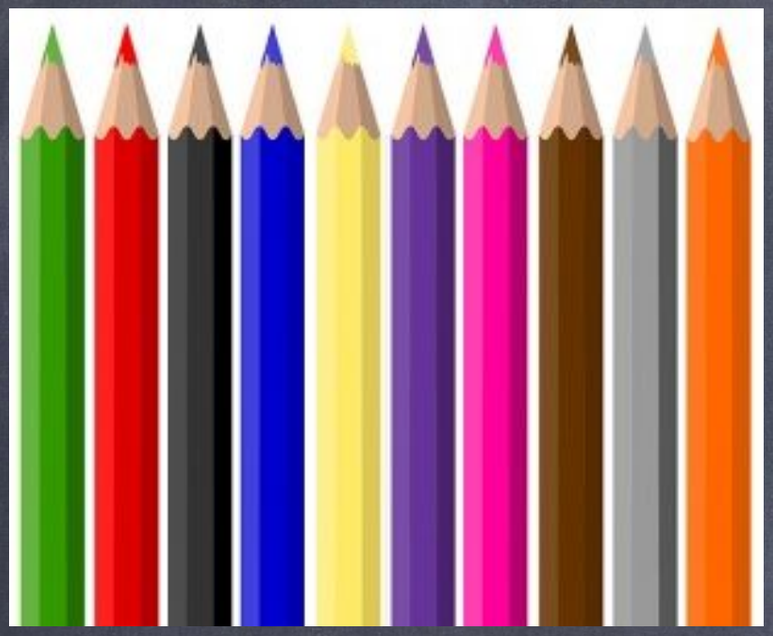
Ordenação por seleção



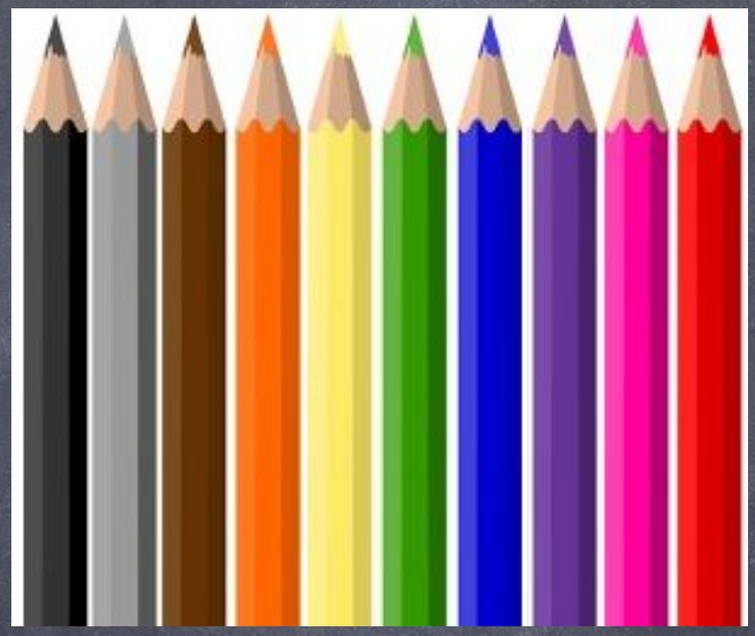


Ordenação por seleção

- E se cada lápis tivesse um número?



6 10 1 7 5 8 9 3 2 4

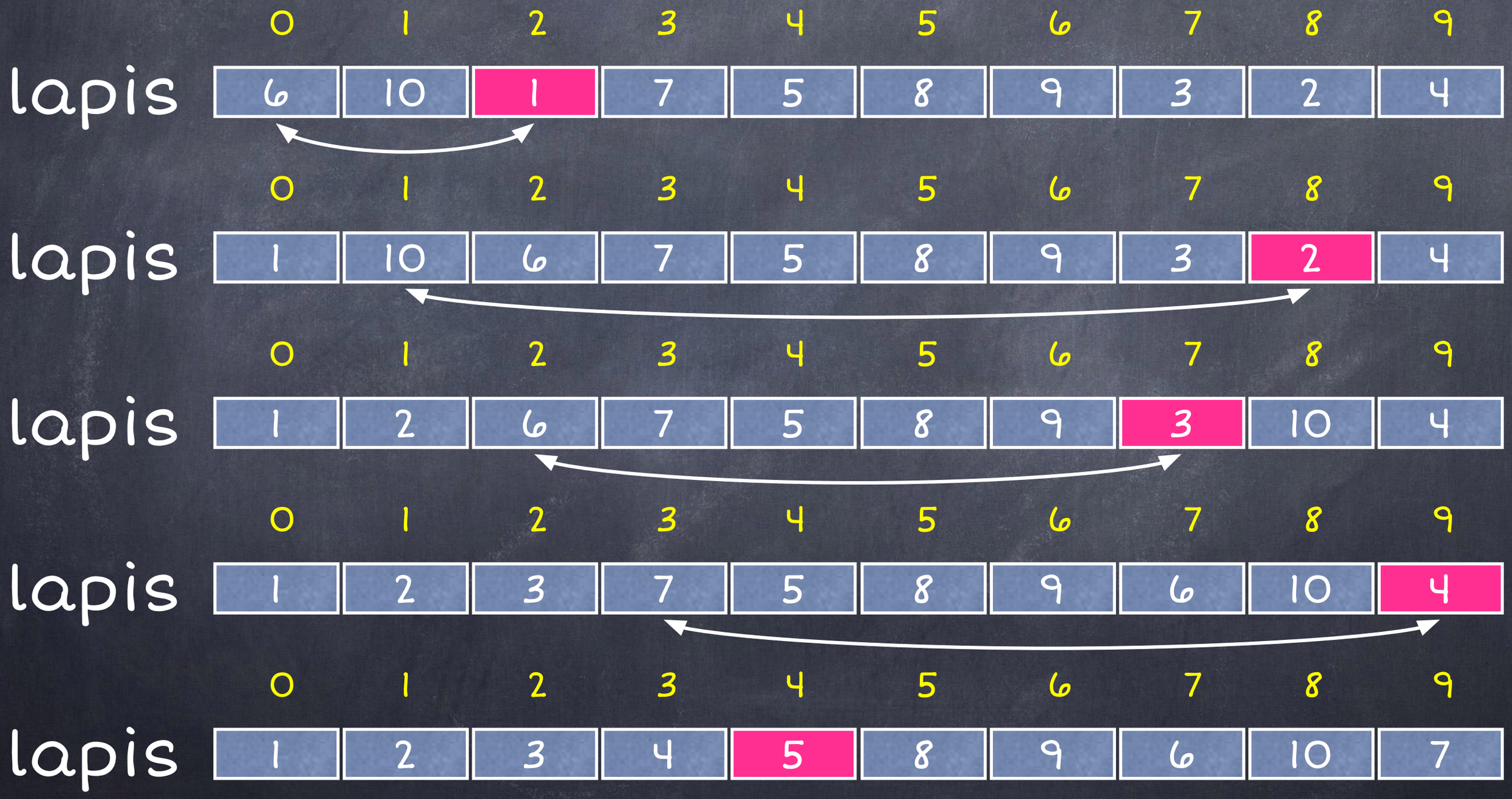


1 2 3 4 5 6 7 8 9 10

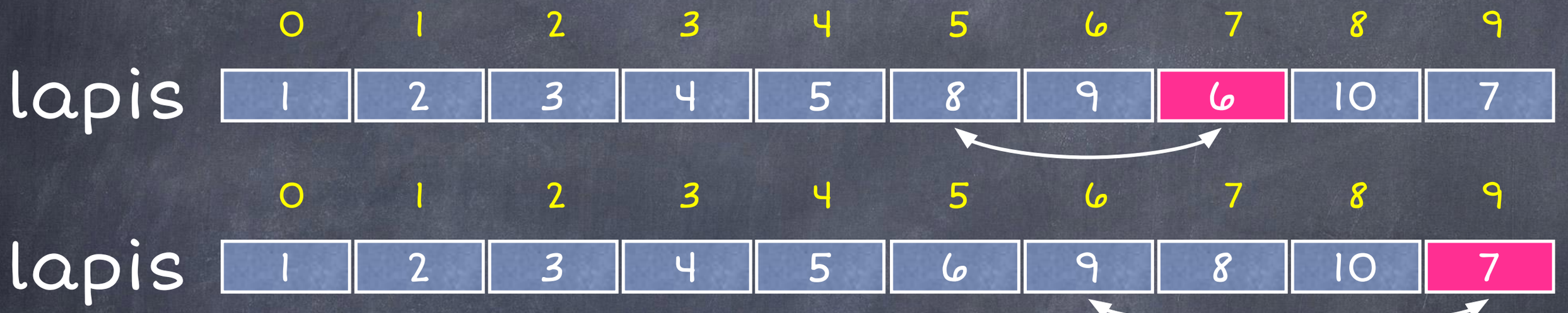
- Podemos guardar em um vetor de inteiros:

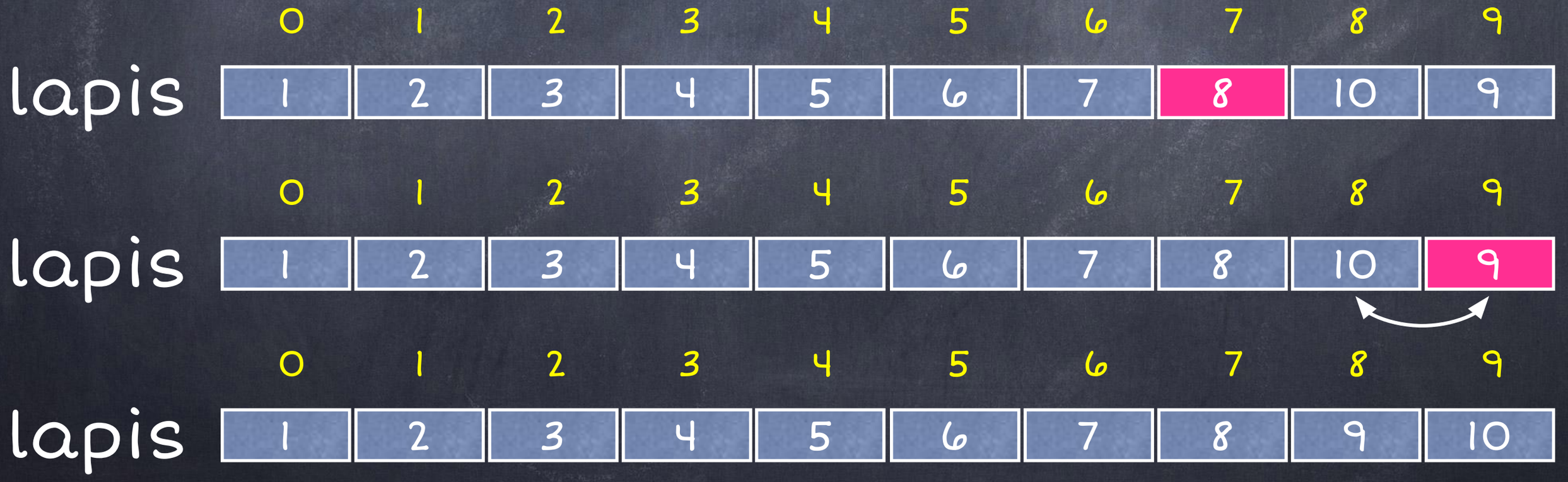
	0	1	2	3	4	5	6	7	8	9
lapis	6	10	1	7	5	8	9	3	2	4

Ordenação por seleção



Ordenação por seleção





Ordenação em C++

- Não precisamos implementar a ordenação toda vez que precisamos usá-la!

- Comando sort é utilizado para ordenar os elementos de um vetor

- Para ordenar um vetor vet com n elementos:

```
sort(vet, vet + n);
```

Ordenação em C++

```cpp
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int vet[100], i, n;

    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &vet[i]);
    }

    sort(vet, vet + n);

    for (i = 0; i < n; i++) {
        printf("%d ", vet[i]);
    }

    printf("\n");

    return 0;
}
```