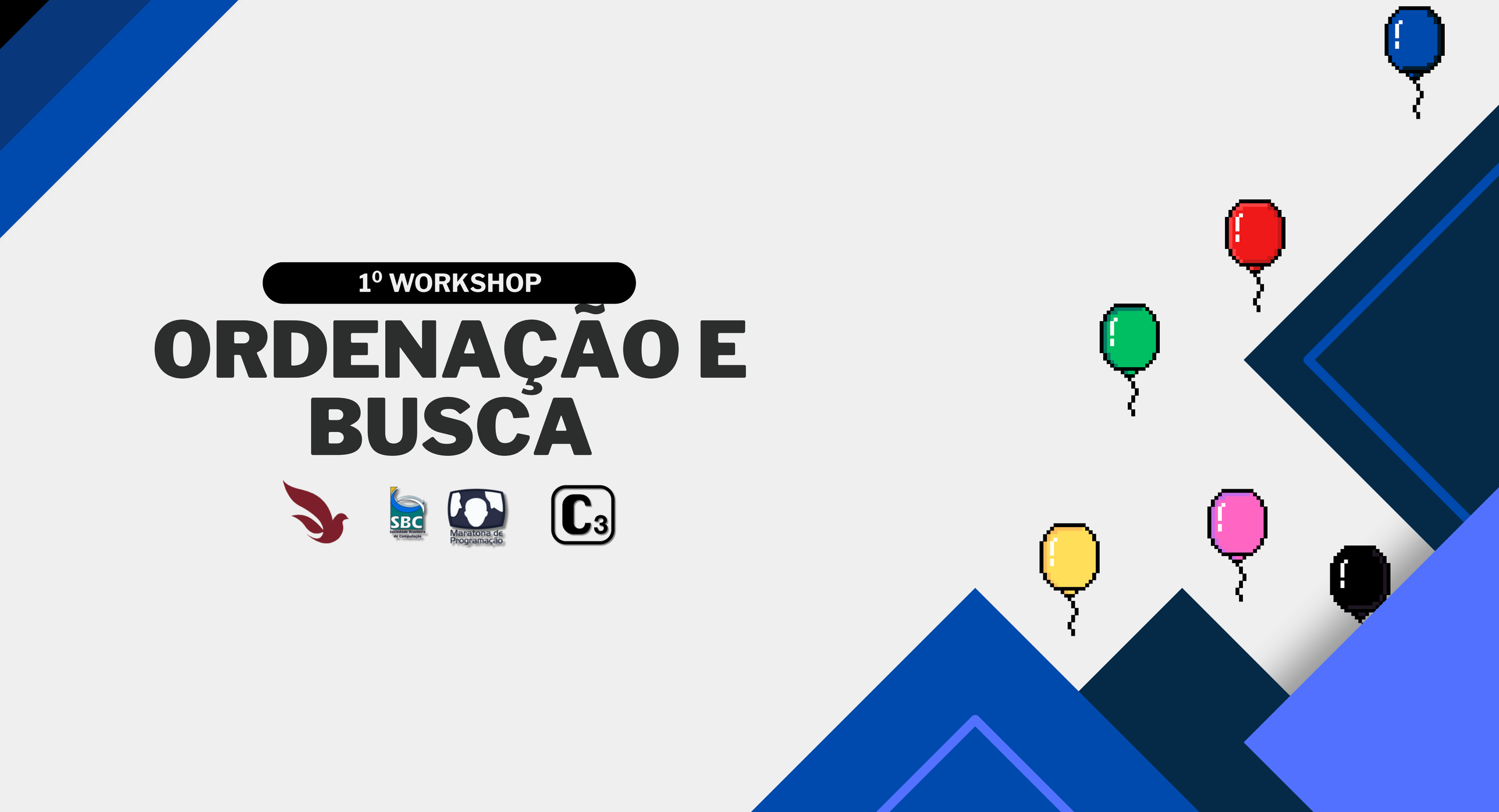
1º WORKSHOP

ORDENAÇÃO E BUSCA



REVISÃO DE AULAS PASSADAS

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main(){

    vector<vector<int> >v(5, vector<int>(4));
    for(int i = 0; i < 5; i++){
        for(int j = 0; j < 4; j++){
            cout << v[i][j] << endl;
        }
    }
}
```

Matriz

REVISÃO DE AULAS PASSADAS

supondo que a matriz é

[2,6,4]
[7,9,8]
[2,3,1]

```
vector<vector<int> >v(3, vector<int>(3));
for(int i = 0; i < 3; i++){
    for(int j = 0; j < 3; j++){
        cout << v[i][j] << endl;
    }
}
```

qual a ordem dos números que será printado?
como fariamos pra printar de outro jeito?

BORA RELEMBRAR ?

beecrowd

1184 - Abaixo da diagonal principal

Ordenação

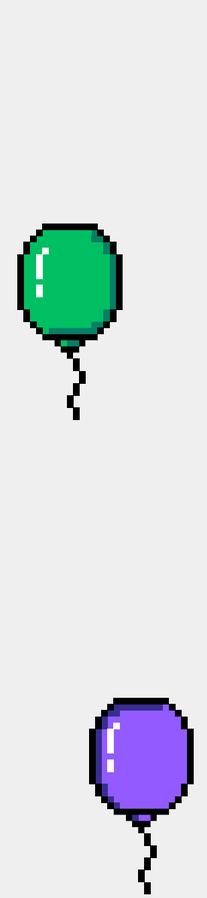
Insertion Sort

```cpp
void InsertionSortVector(vector<int> &arr){
    for (int i = 1; i < arr.size(); i++){
        int key = arr[i];
        int j = i;
        while (j > 0 && arr[j-1] > key){
            arr[j] = arr[j-1];
            j--;
        }
        arr[j] = key;
    }
}
```

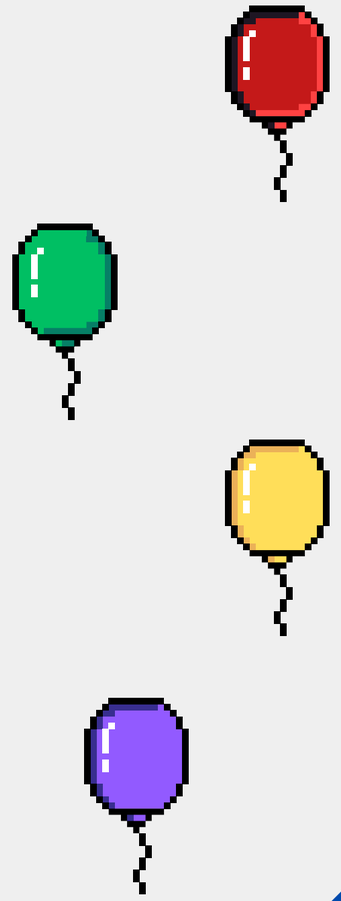
BubbleSort

```
void bubbleSort(vector<int> &arr){

    for (int i = 0; i < arr.size()-1; i++){
        for (int j = 0; j < arr.size()-i-1; j+
            if (arr[j] > arr[j+1]){
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
```



Busca



O QUE É UMA BUSCA?

Para busca, definimos o conceito de key e value.

⟶ A key, seria o ínidice que está localizado o value no vetor.

⟶ é justamente a localização do value que estamos querendo.

se value = 20

[10,40, 70, 20]

key = ? 3

TIPOS DE BUSCA

Busca sequencial

- Forma mais simples de busca
- Considerando que o vetor tem **n** elementos, a complexidade desse algoritmo é O(n)

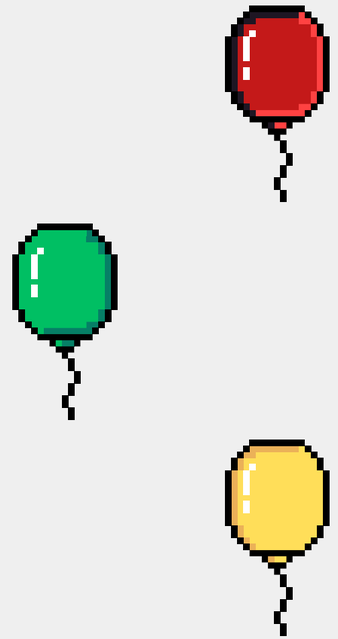
Busca sequencial

```cpp
int main(){

    vector<int>v(5);
    v = {10,90,40,50,20};

    int value = 40;
    int key = -1;
    for(int i = 0; i < 5; i++){
        if(v[i] == value){
            key = i;
            break;
        }
    }

    cout << key << endl;
}
```



TIPOS DE BUSCA

Busca binária

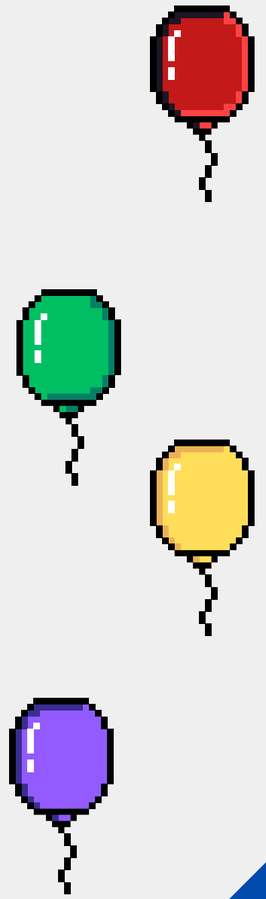
- Critério : vetor agora está obrigatoriamente ordenado
- Sua complexidade agora é O(log n)

Busca binária

O que nós podemos ter de vantagem pelo vetor ser ordenado?

ex: value = 30
key = ?

[5 , 10 , 20 , 25 , 27 , 30 , 32 , 40]



Busca binária

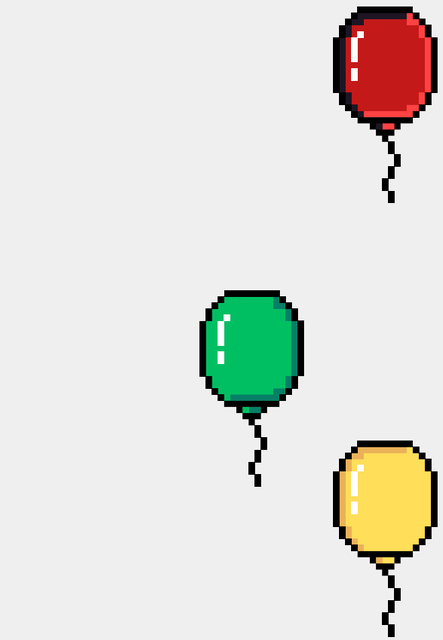
[5 , 10 , 20 , 25 , 27 , 30 , 32 , 40]

Posições

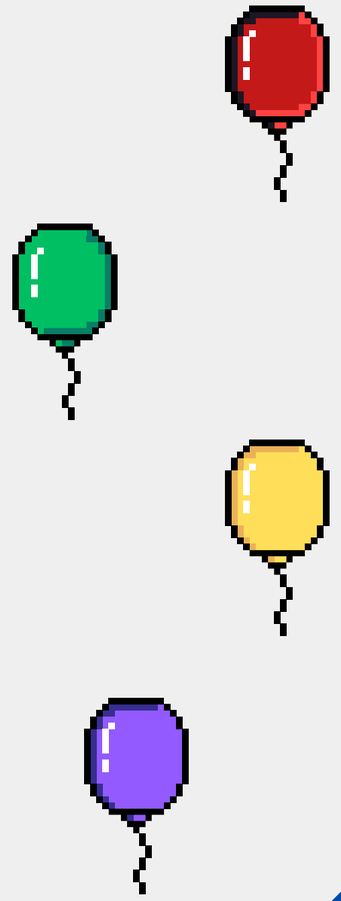
low = 0
high = 7

$$mid = \frac{(low + high)}{2}$$

enquanto low <= high



BORA PRATICAR?



beecrowd

Binary search