Exercicios Resolvidos - Vetores

Programacao Estruturada

Prof. André del Mestre

Tecnicas uteis na programaçao com vetores

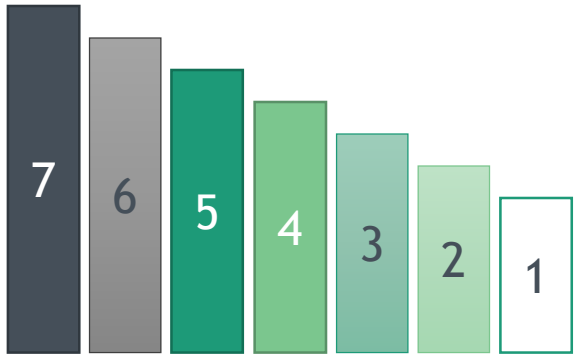
Tecnicas uteis em vetores

Parametrize o tamanho utilizando #define


```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Tecnicas uteis em vetores

Complemento do indice

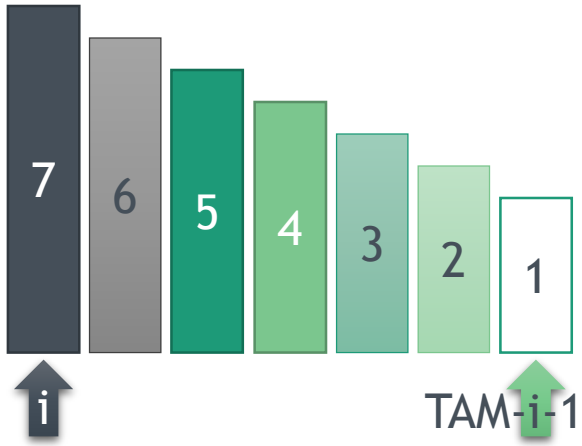


Vetor x[] preenchido com complemento do indice

```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Tecnicas uteis em vetores

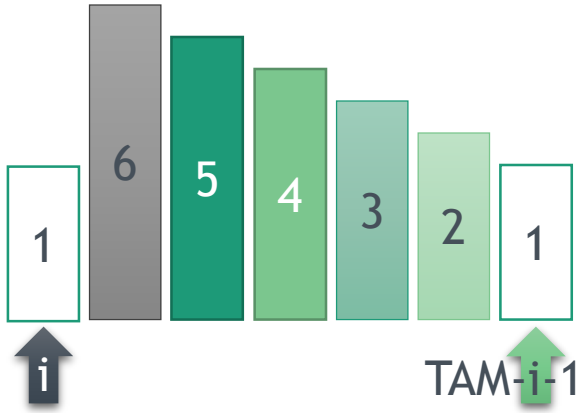
Troca de valor entre duas posiçoes de um mesmo vetor




```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Tecnicas uteis em vetores

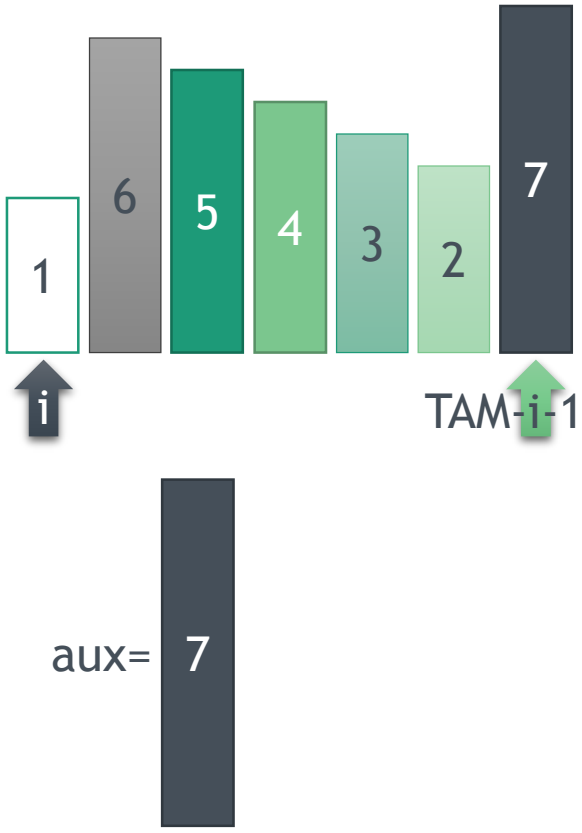
Troca de valor entre duas posiçoes de um mesmo vetor




```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Tecnicas uteis em vetores

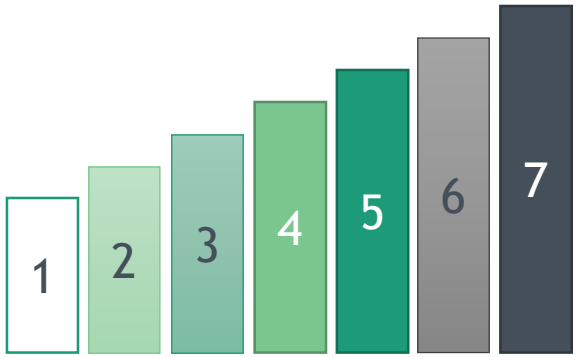
Troca de valor entre duas posiçoes de um mesmo vetor




```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Tecnicas uteis em vetores

Imprima o vetor para fins de debug




```
x[0] = 1
x[1] = 2
x[2] = 3
x[3] = 4
x[4] = 5
x[5] = 6
x[6] = 7
```

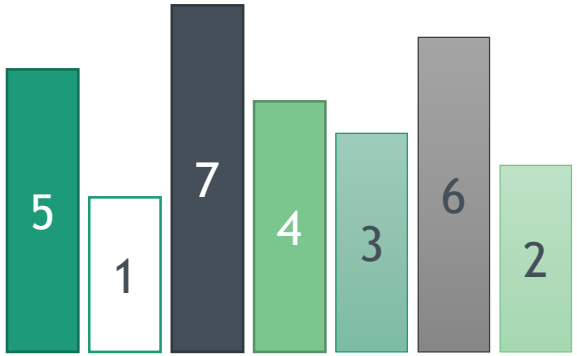
```c
#define TAM 7
int main (){
  int i, x[TAM], num;
  for(i=0; i<TAM; i++){
     x[i]=TAM-i;
  }
  ...
  for(i=0; i<TAM/2; i++){
    aux=x[i];
    x[i]=x[TAM-i-1];
    x[TAM-i-1]=aux;
  }
  ...
  for(i=0; i<TAM; i++){
    printf("x[%i] = %i\n", i, x[i]);
  }
```

Ex. 2 – Busca em Vetor

Busca em vetor

Quando um elemento esta ordenado, não se mexe mais

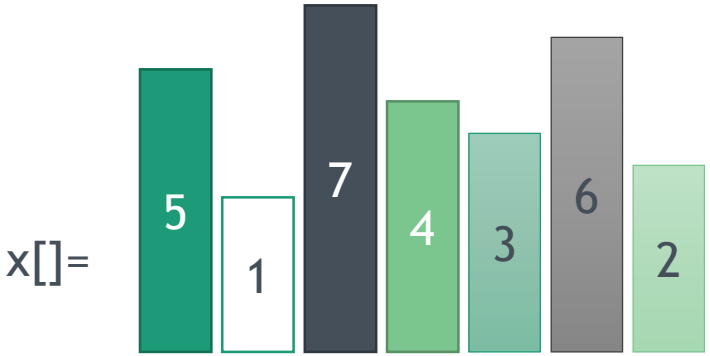
x[]=




```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

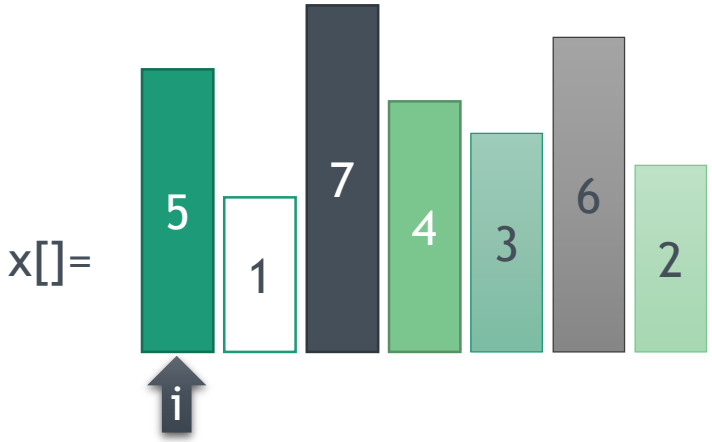
Busca o indice da primeira ocorrencia de um numero em um vetor




```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

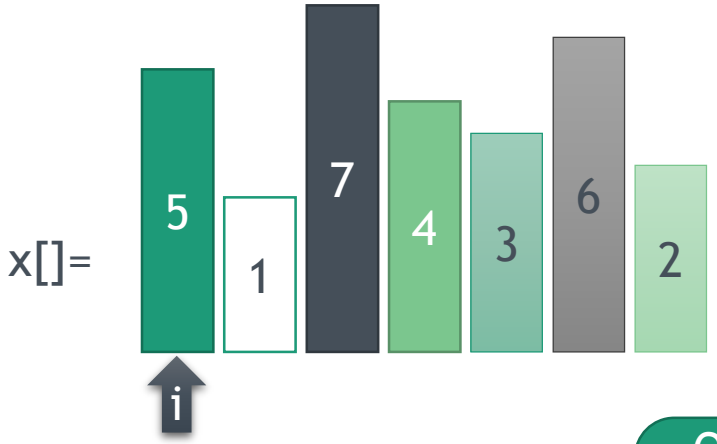
Busca o indice da primeira ocorrencia de um numero em um vetor




```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

Busca o indice da primeira ocorrencia de um numero em um vetor



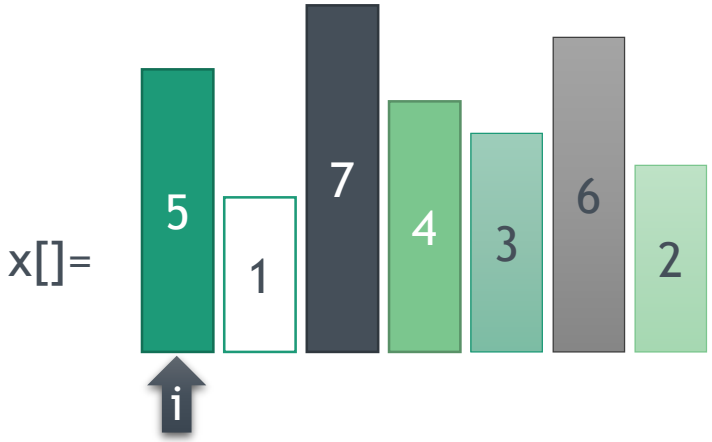

```
for(i=0; i<TAMANHO; i++){
     x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
     i++;
}
```

A busca nao vai ultrapassar tamanho do vetor

Quando encontar um numero igual ao digitado, sai do laço while

Busca em vetor

Busca o indice da primeira ocorrencia de um numero em um vetor

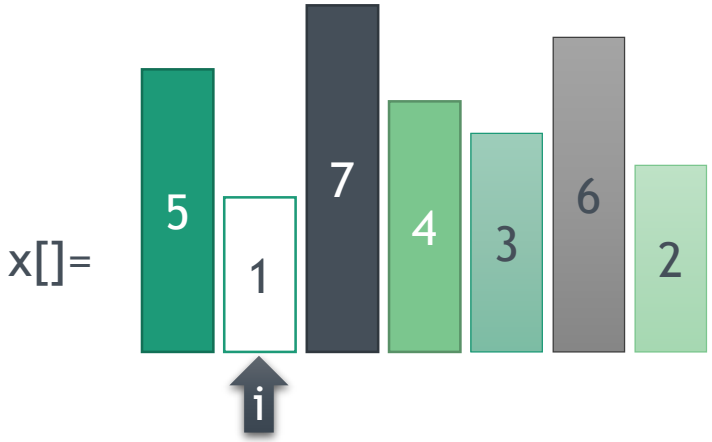


Verdadeiro!

```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

Busca o indice da primeira ocorrencia de um numero em um vetor

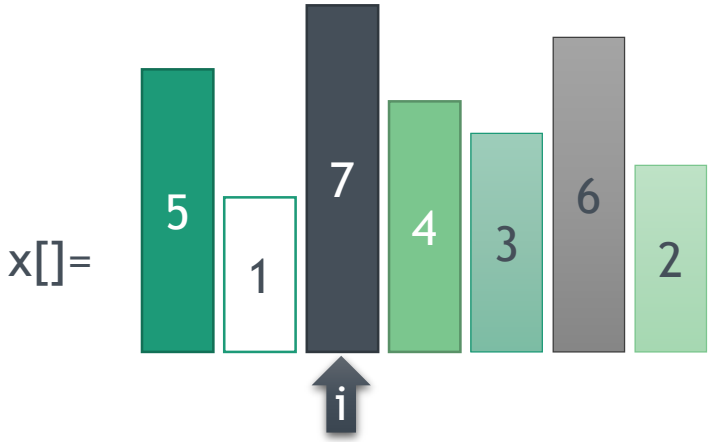


Verdadeiro!

```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

Busca o indice da primeira ocorrencia de um numero em um vetor

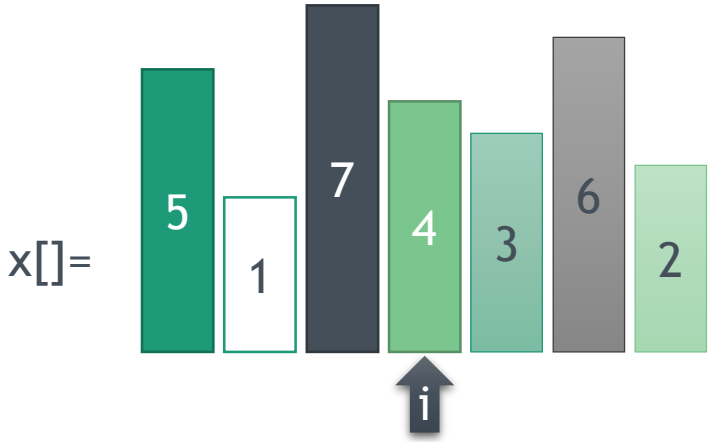


Verdadeiro!

```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

Busca em vetor

Busca o indice da primeira ocorrencia de um numero em um vetor




```
for(i=0; i<TAMANHO; i++){
    x[i] = rand()%TAMANHO;
}
scanf("%i", &num);
i=0;
while(x[i]!=num && i<TAMANHO){
    i++;
}
```

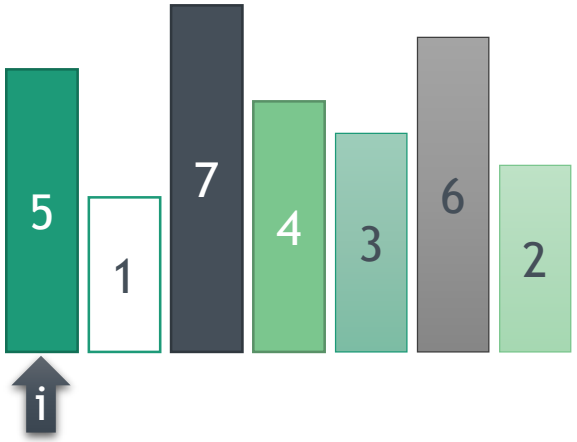
Falso!

num encontrado em **x[3]**!

Ex. 3 - Ordena Vetor

Ordenação de vetor

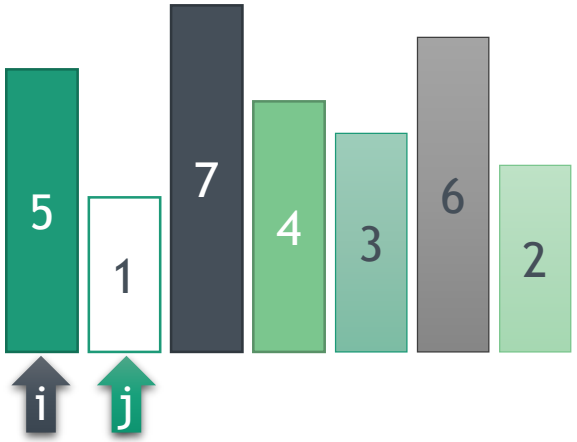
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

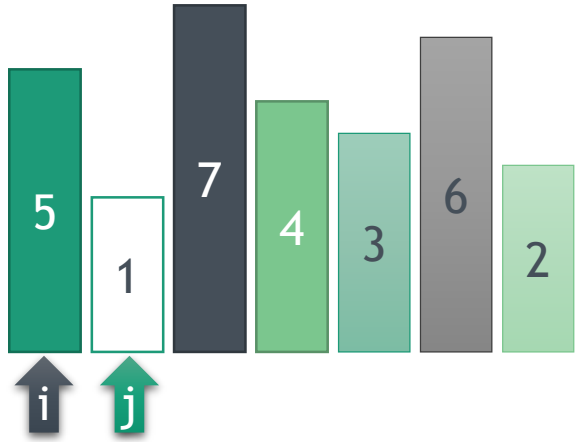
Quando um elemento esta ordenado, não se mexe mais




```
    for(i=0; i<TAMANHO-1; i++){
        for(j=i+1; j<TAMANHO; j++){
            if(x[i]>x[j]){
                aux=x[i];
                x[i]=x[j];
                x[j]=aux;
            }
        }
    }
```

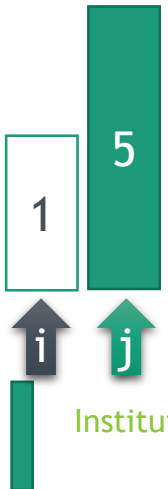
Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



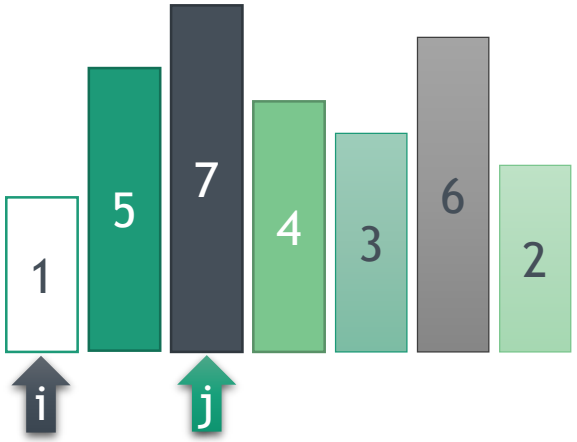

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!



Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

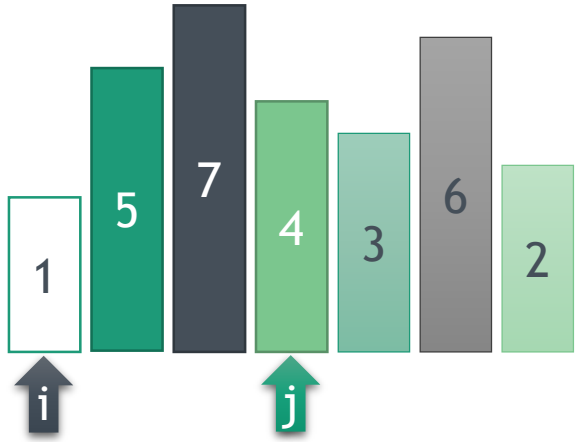


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

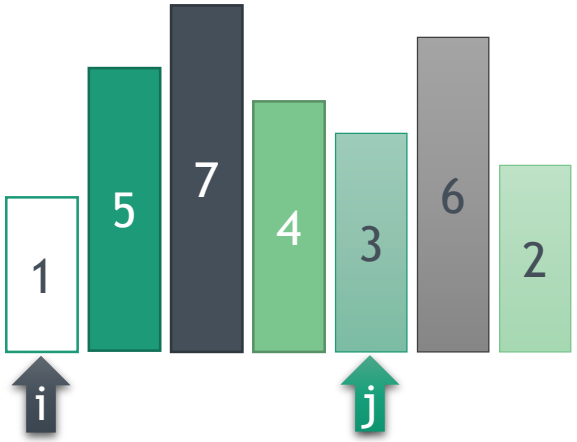


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

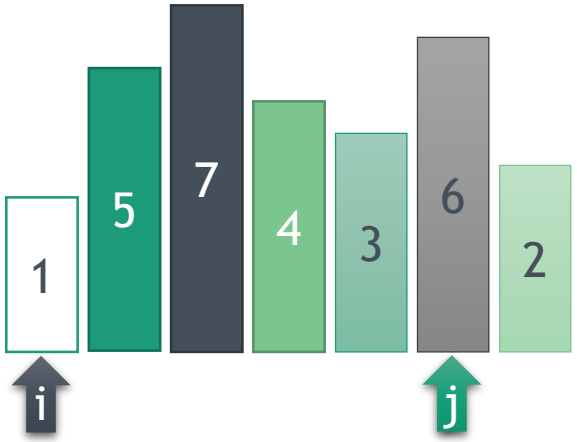


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

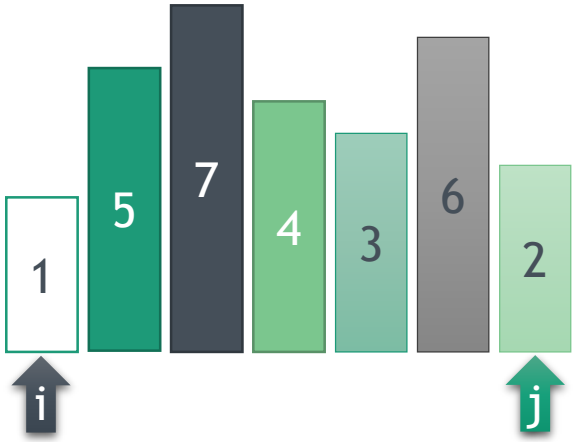


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

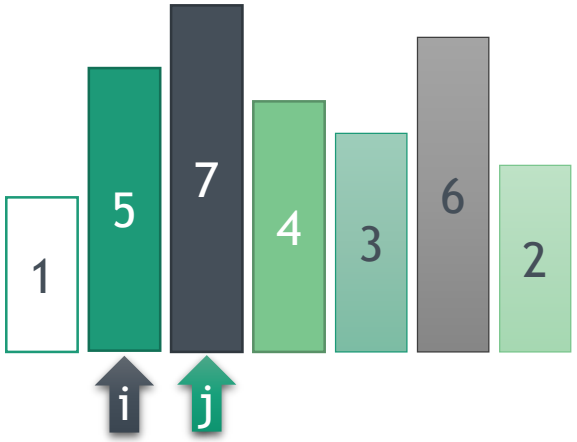


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

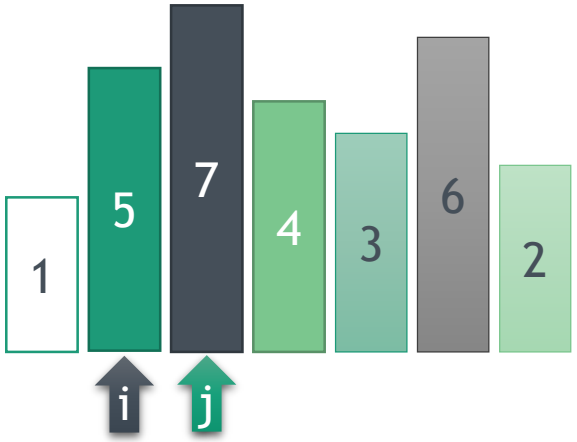
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

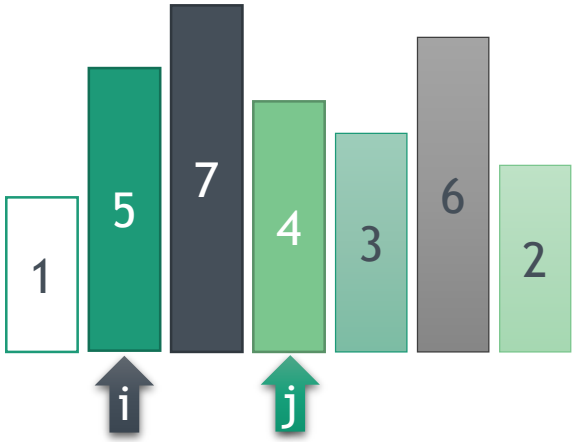


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

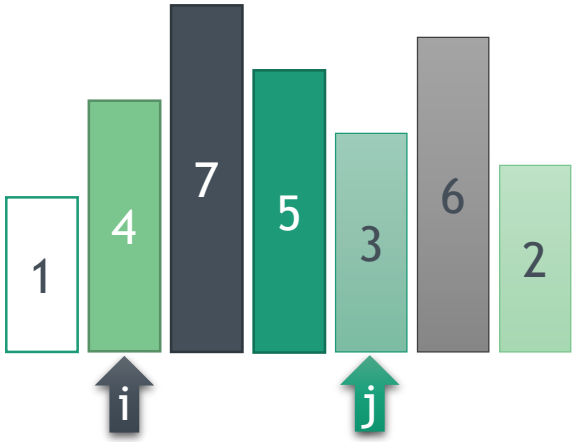



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

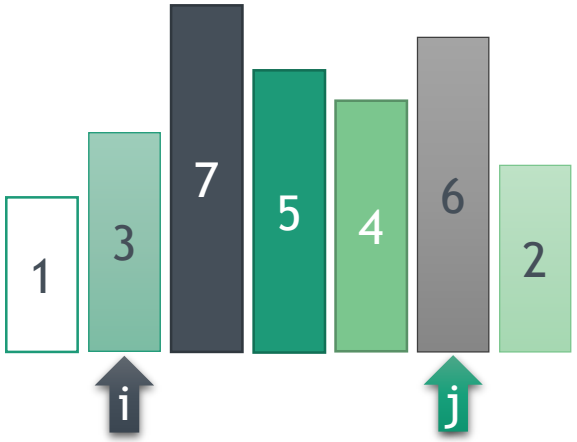



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

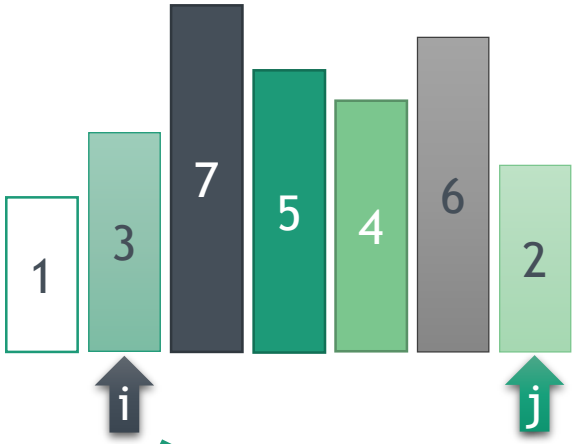


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

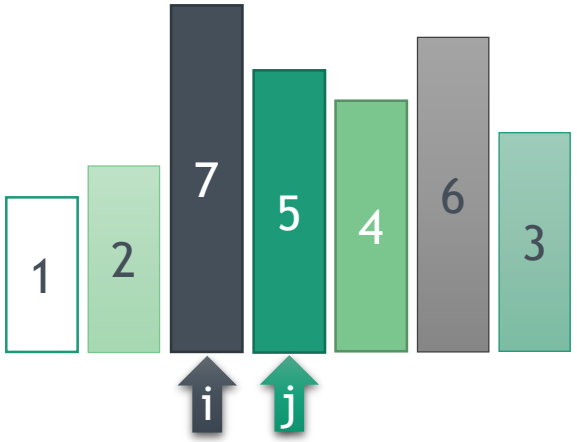



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

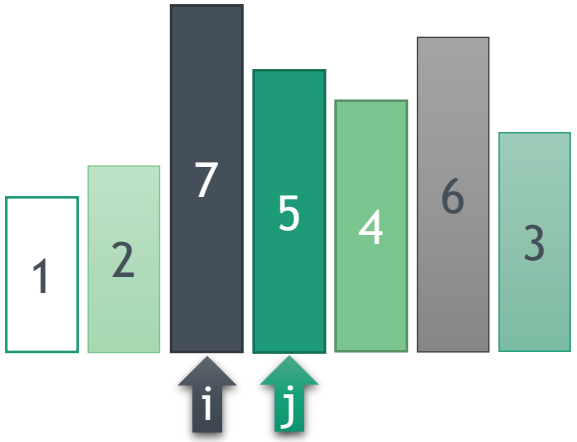
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

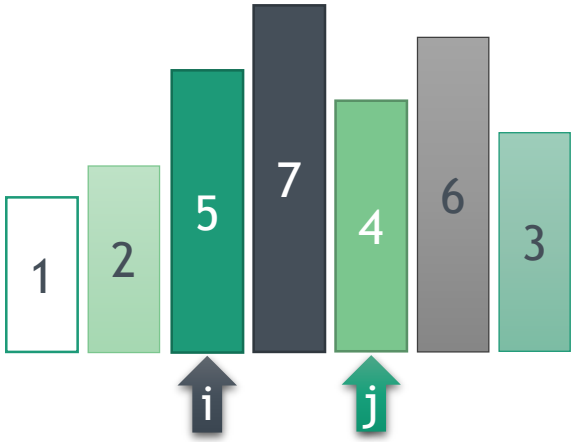


Verdadeiro!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

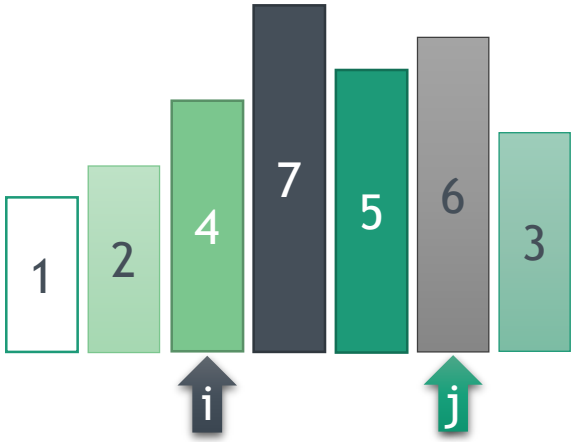



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

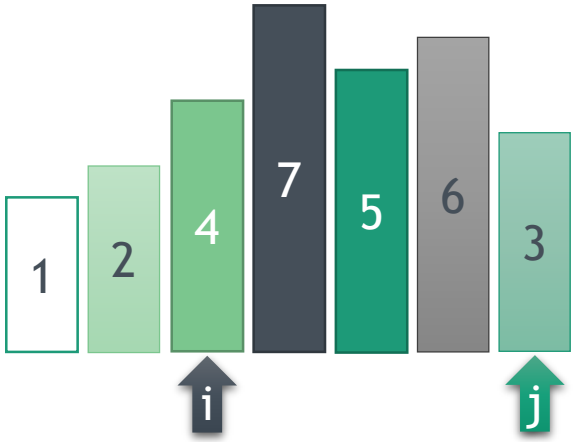


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

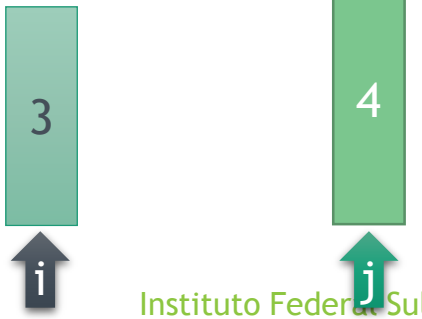
Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



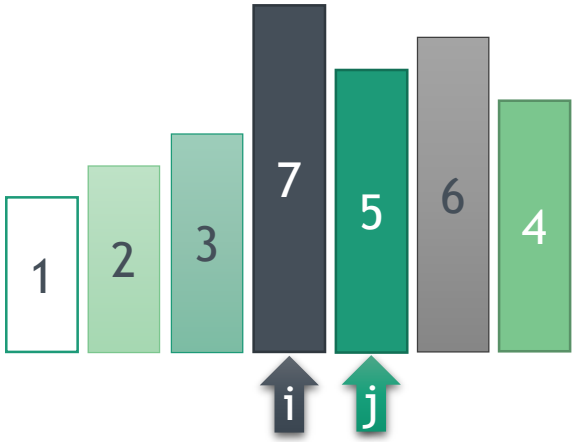

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!



Ordenação de vetor

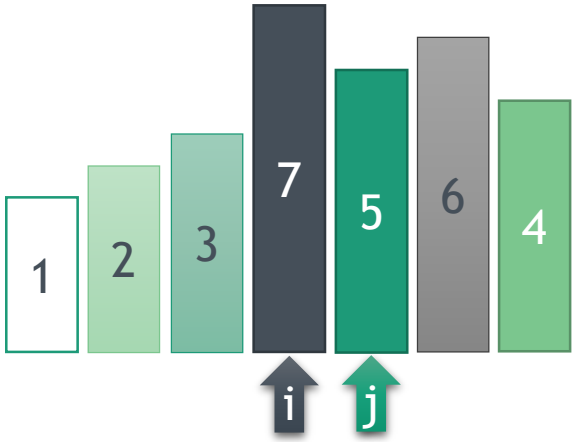
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

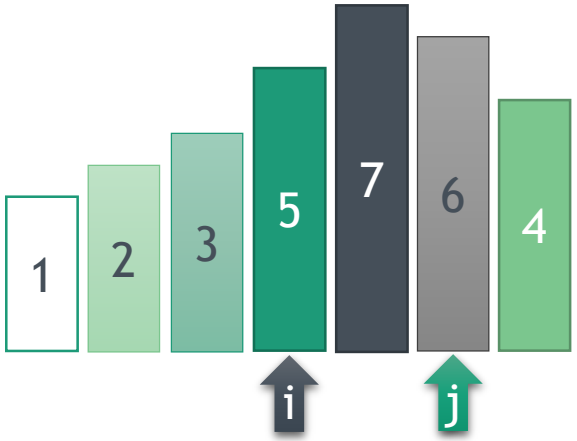



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

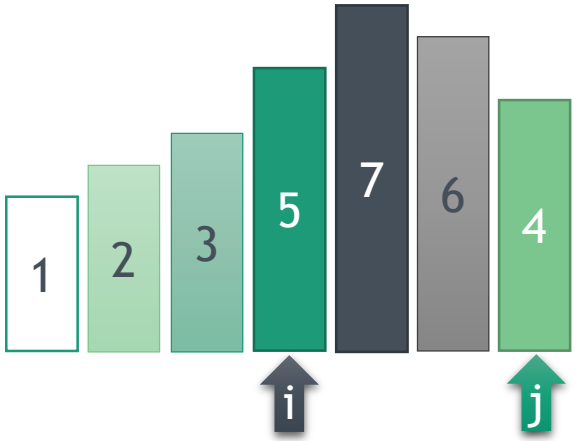


Falso!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

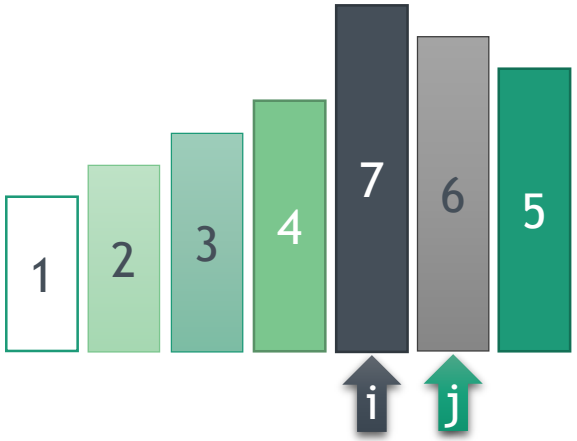



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

Ordenação de vetor

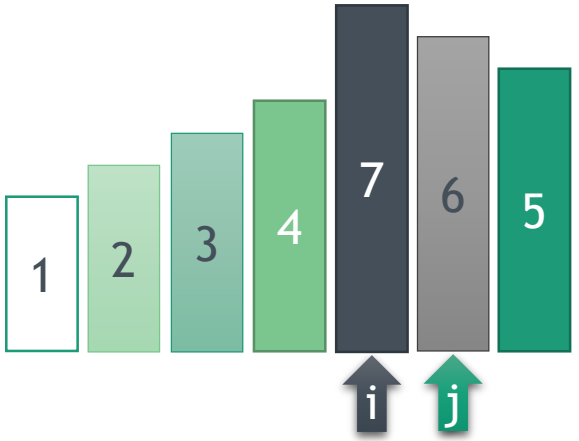
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais

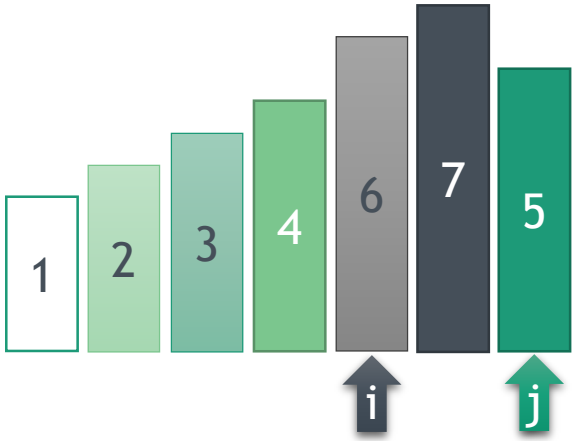



```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!

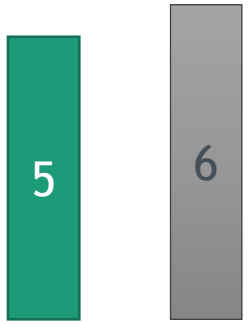
Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



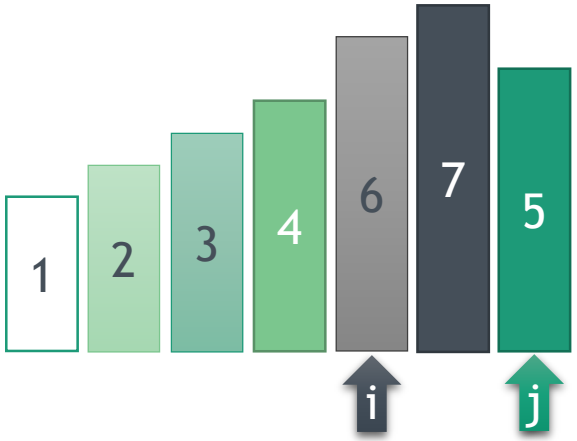

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!



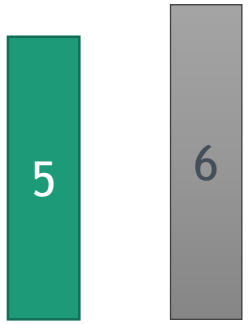
Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



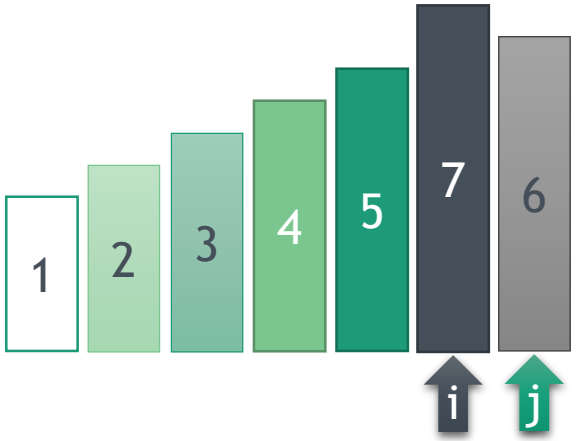

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!



Ordenação de vetor

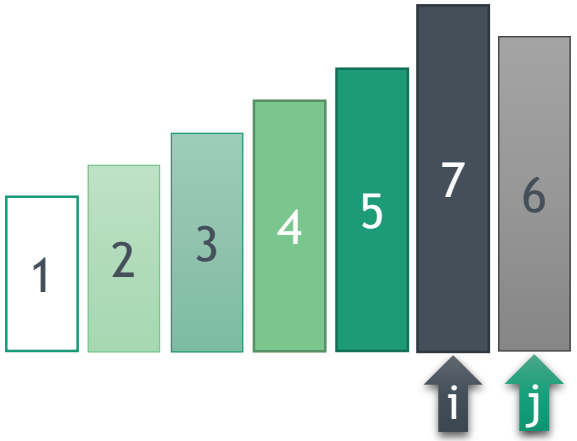
Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

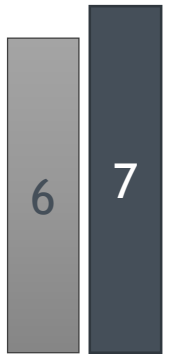
Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



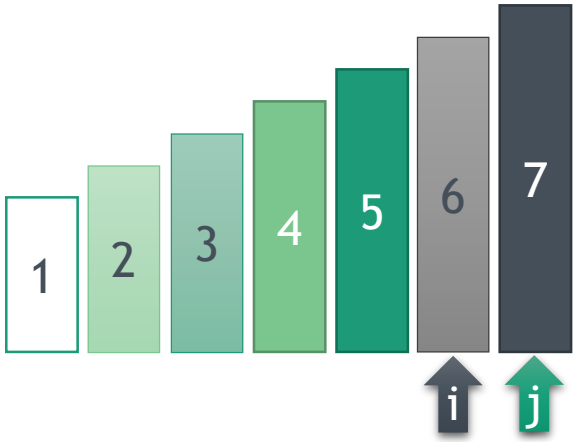

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Verdadeiro!



Ordenação de vetor

Quando um elemento esta ordenado, não se mexe mais



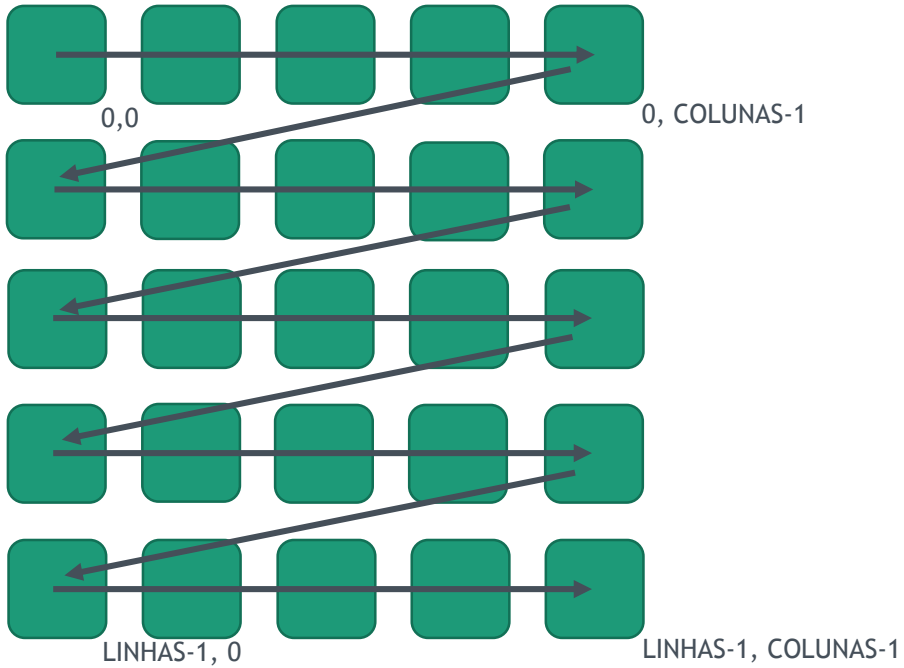
ALELUIA!

```
for(i=0; i<TAMANHO-1; i++){
    for(j=i+1; j<TAMANHO; j++){
        if(x[i]>x[j]){
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
        }
    }
}
```

Matrizes - preenchimento

Preenchendo matrizes

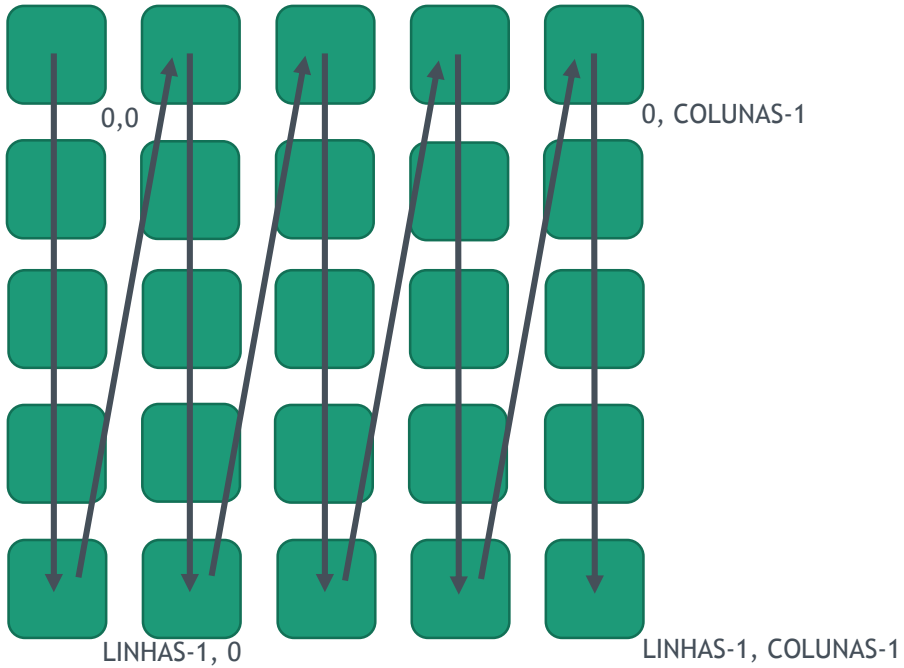
A ordem de preenchimento importa!




```
cont=0;
for(i=0; i<LINHAS; i++){
    for(j=0; j<COLUNAS; j++){
        m[i][j] = cont;
        cont++;
    }
}
```

Preenchendo matrizes

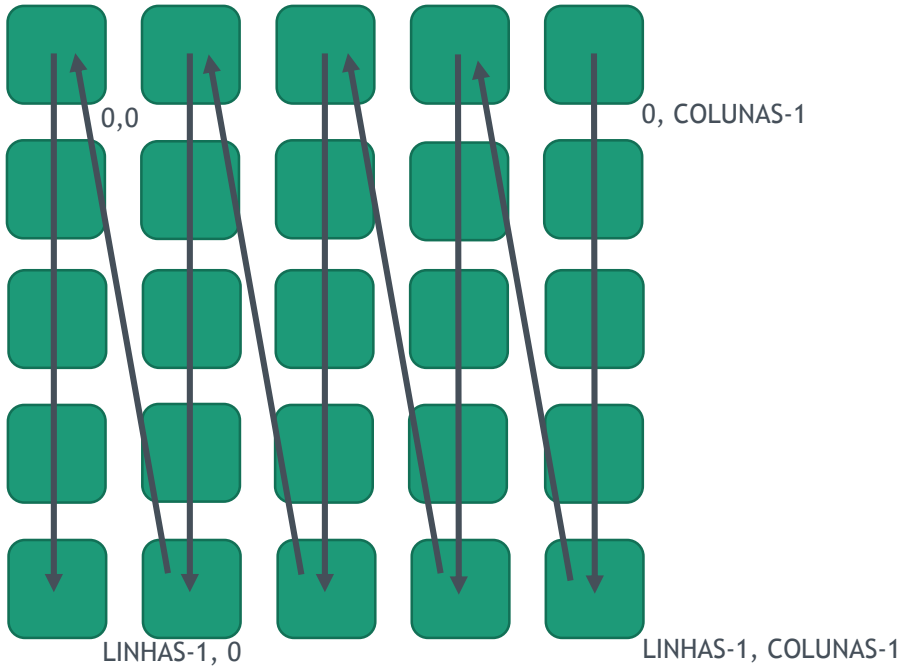
A ordem de preenchimento importa!




```
cont=0;
for(j=0; j<COLUNAS; j++){
    for(i=0; i<LINHAS; i++){
        m[i][j] = cont;
        cont++;
    }
}
```

Preenchendo matrizes

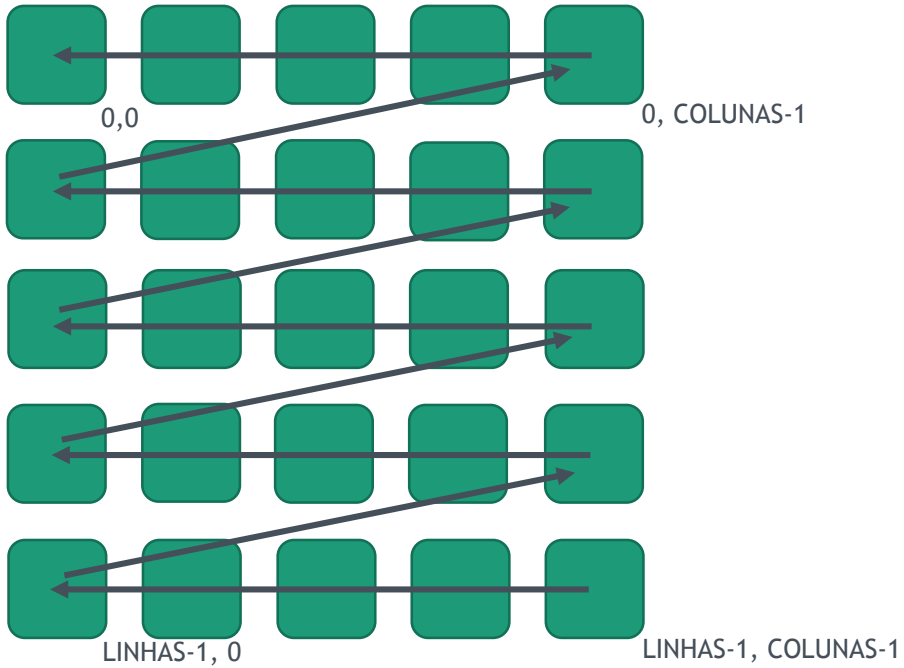
A ordem de preenchimento importa!




```
//Complete o codigo para
//reproduzir o desenho
cont=0;
for(;;){
    for(;;){
        m[][] = cont;
        cont++;
    }
}
```

Preenchendo matrizes

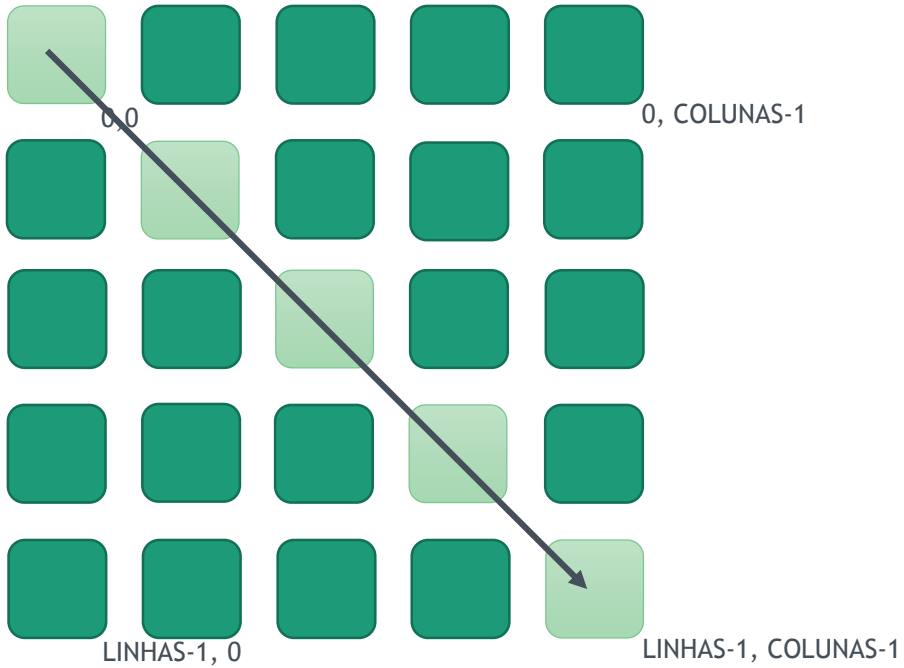
A ordem de preenchimento importa!




```
//Complete o codigo para
//reproduzir o desenho
cont=0;
for(;;){
    for(;;){
        m[][] = cont;
        cont++;
    }
}
```

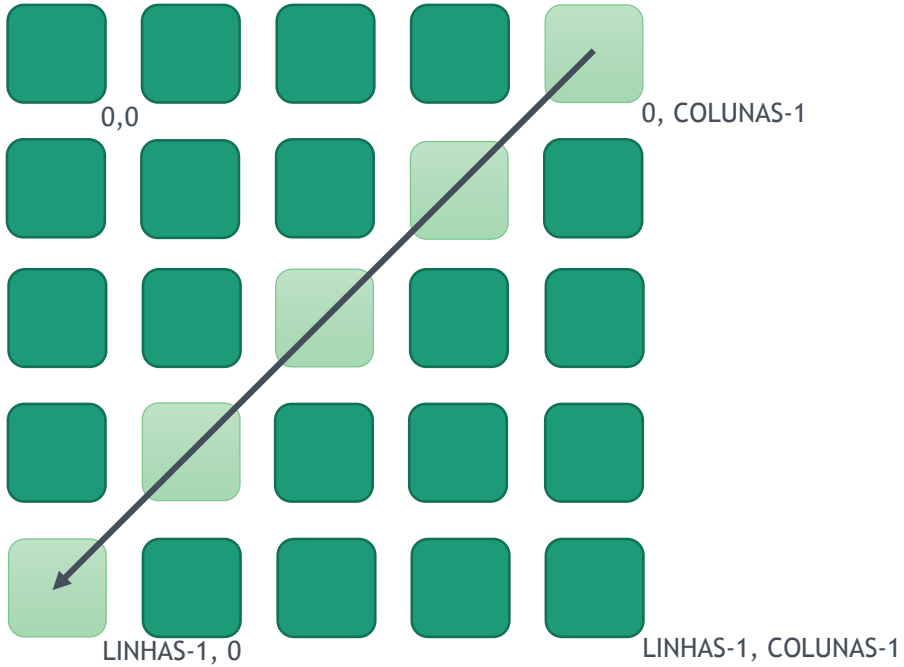
Matrizes - diagonais

Percorrendo diagonais




```
scanf("%i", &num);
i=0;
while(i<LINHAS){
    m[i][i] = num;
    i++;
}
```

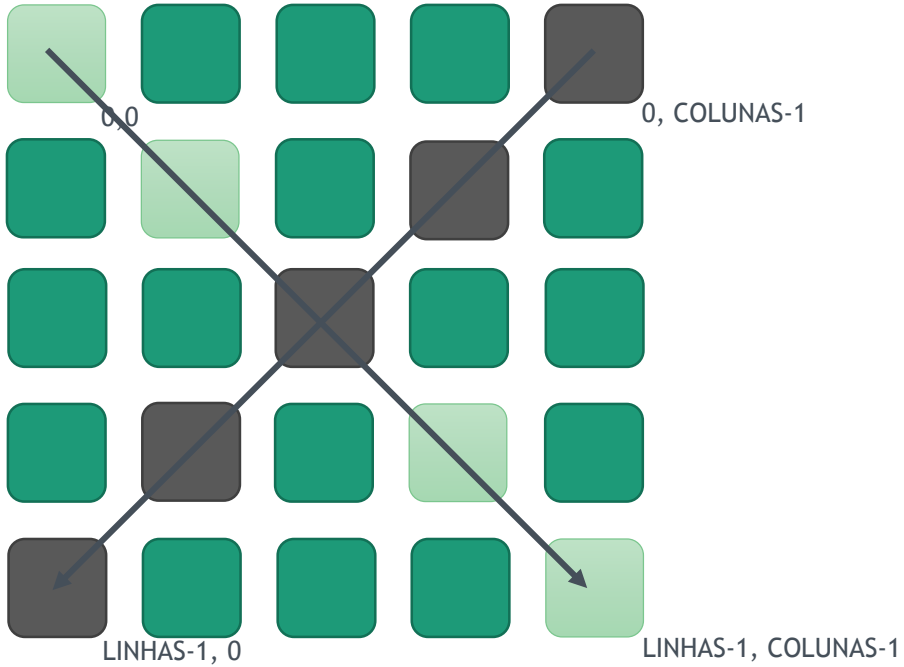
Percorrendo diagonais




```
scanf("%i", &num);
i=0; j=COLUNAS-1;
while(i<LINHAS){
    m[i][j] = num;
    i++; j--;
}
```

Busca em vetor

Quando um elemento esta ordenado, não se mexe mais




```
for(i=0; i<LINHAS; i++){
    principal[i] = m[i][i];
    secundaria[i] = m[LINHAS-i-1][i];
}
```

Tecnica do COMPLEMENTO

MUITO OBRIGADO

Prof. André del Mestre

www.ifsul.edu.br
almmartins@charqueadas.ifsul.edu.br