make history.

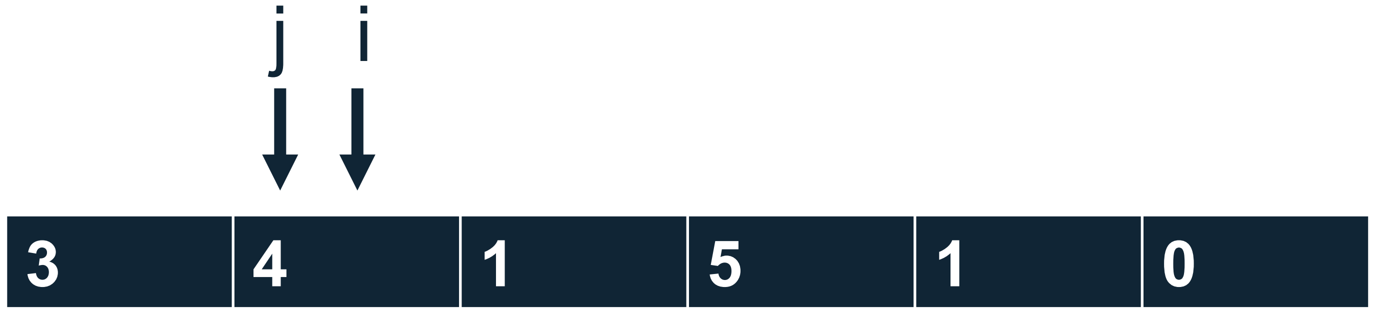
THE UNIVERSITY of ADELAIDE

Insertion sort

Dr. Anna Kalenkova

Insertion sort

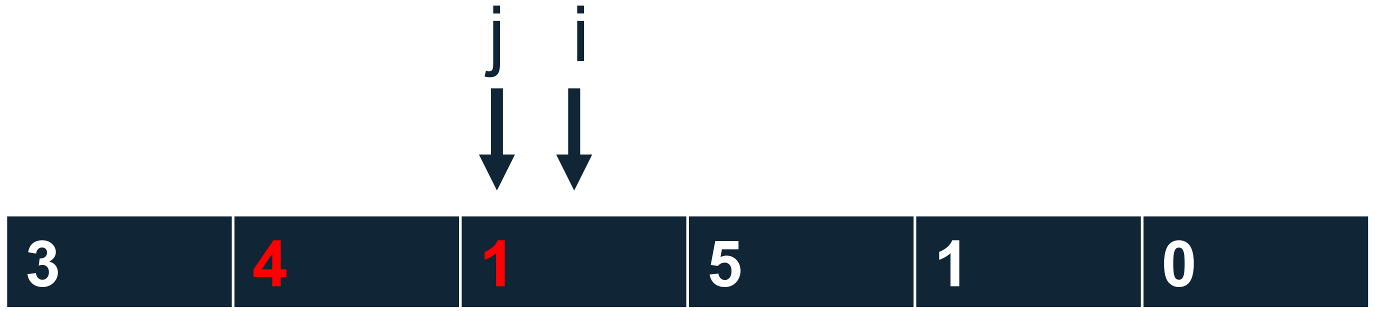
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



3	4	1	5	1	0

Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```

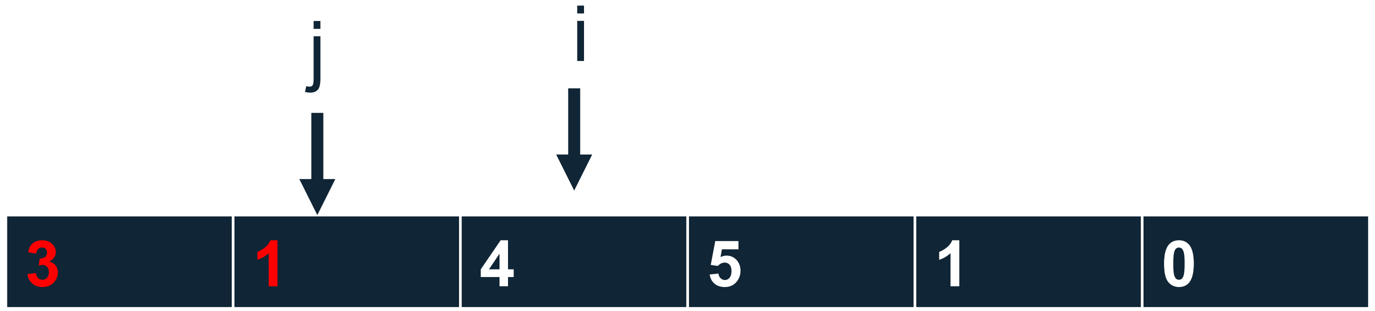


j i

3	4	1	5	1	0

Insertion sort

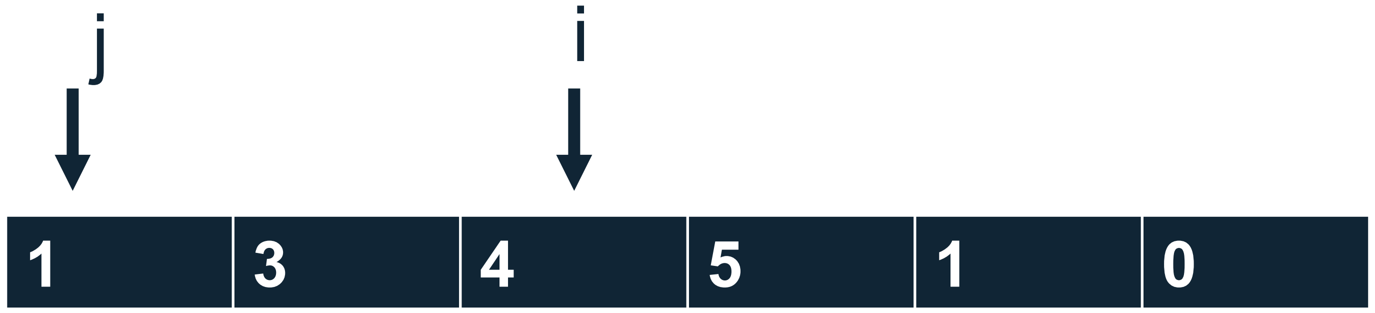
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



3	1	4	5	1	0

Insertion sort

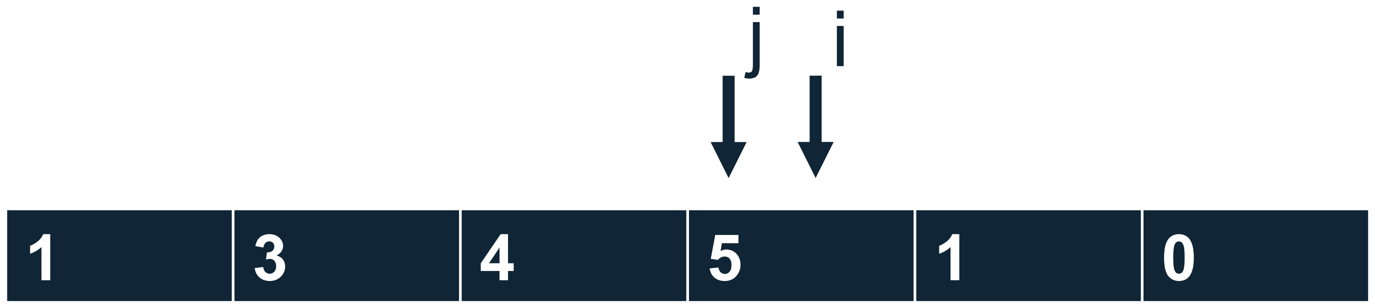
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
               if(array.at(j) < array.at(j-1)) {
                    // swap array.at(j+1) and array.at(j)
               }
               else break;
          }
     }
}
```



1	3	4	5	1	0

Insertion sort

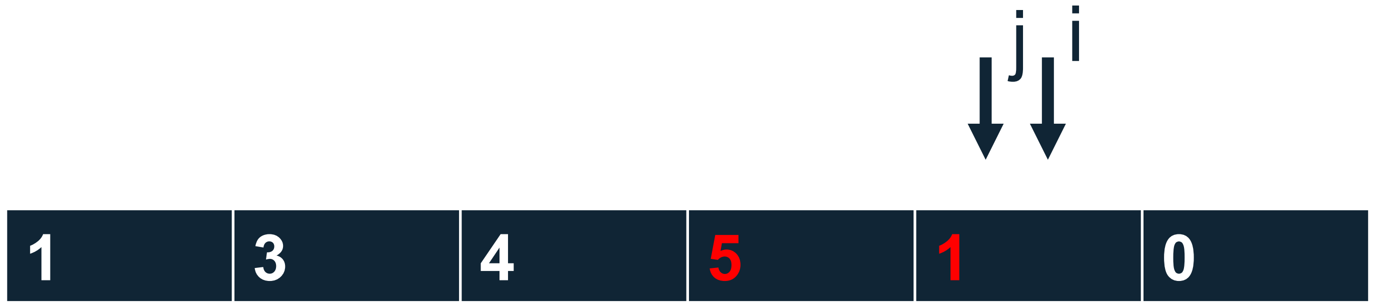
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



1	3	4	5	1	0

Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
               if(array.at(j) < array.at(j-1)) {
                    // swap array.at(j+1) and array.at(j)
               }
               else break;
          }
     }
}
```

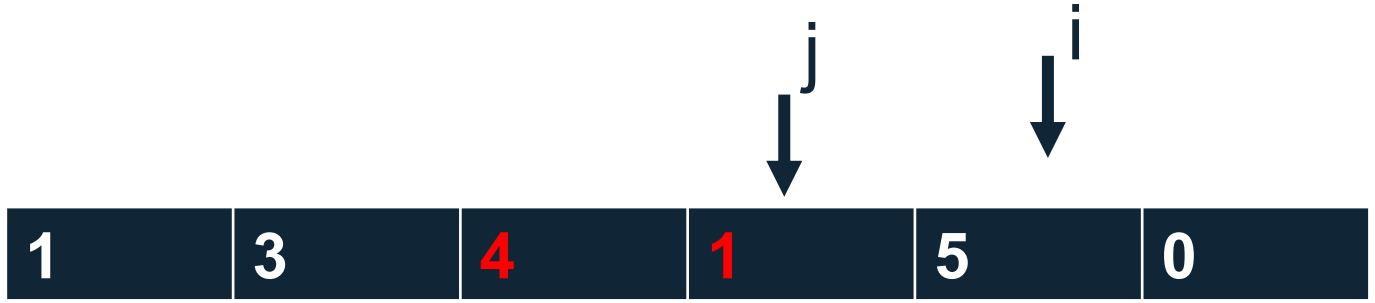


j i

1	3	4	5	1	0

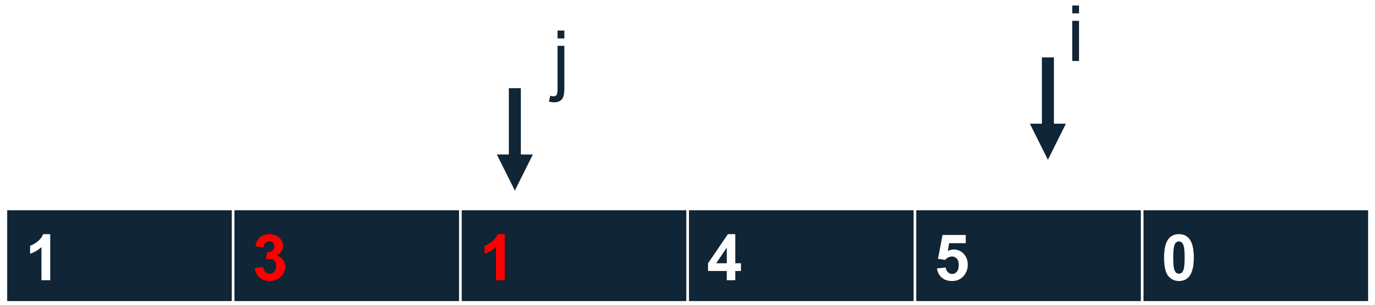
Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



Insertion sort

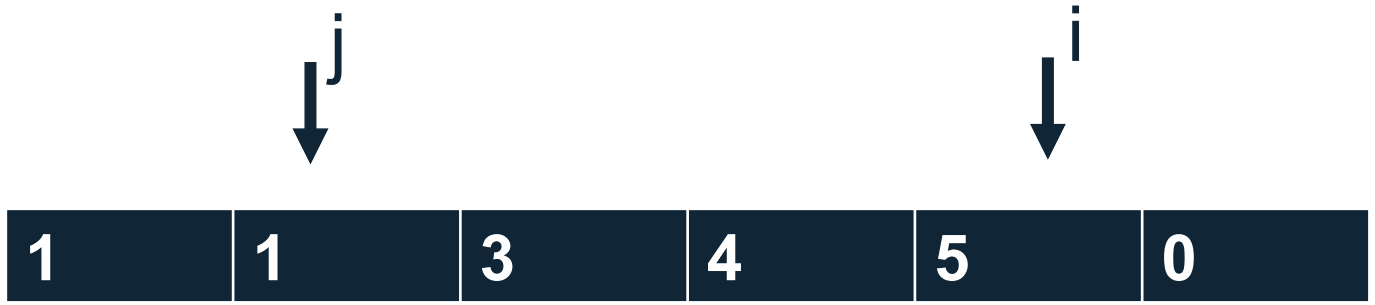
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



1	3	1	4	5	0

Insertion sort

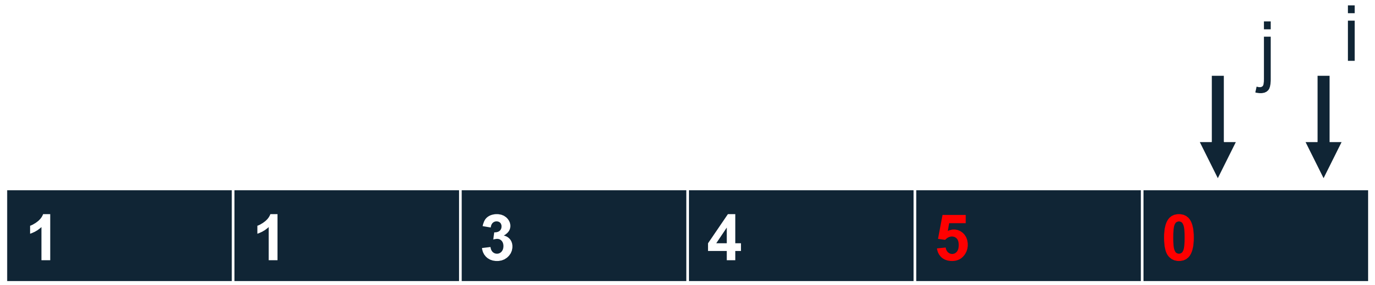
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



1	1	3	4	5	0

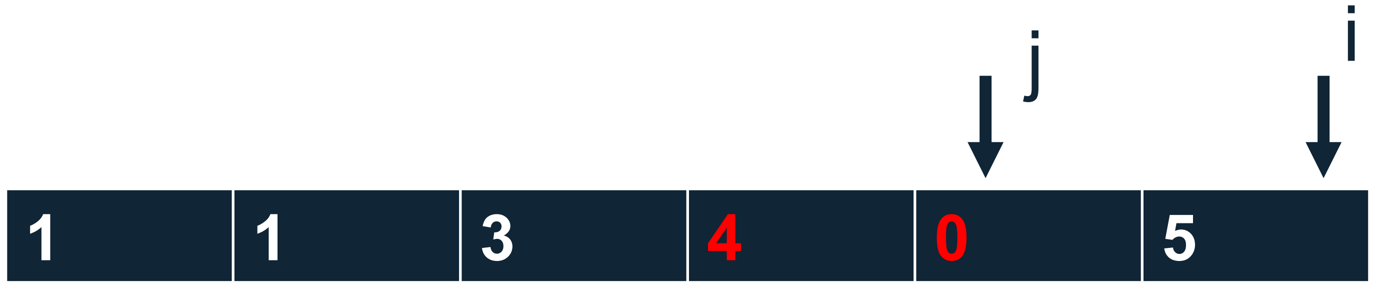
Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
               if(array.at(j) < array.at(j-1)) {
                    // swap array.at(j+1) and array.at(j)
               }
               else break;
          }
     }
}
```



Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



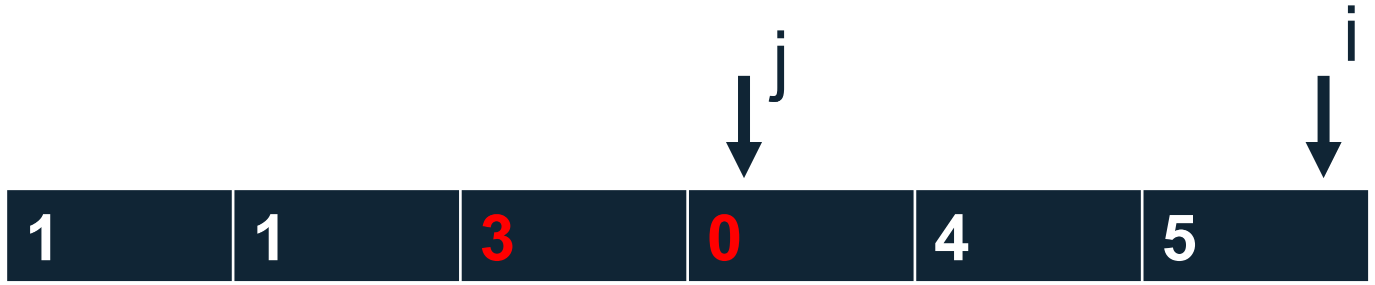
j

i

1	1	3	4	0	5

Insertion sort

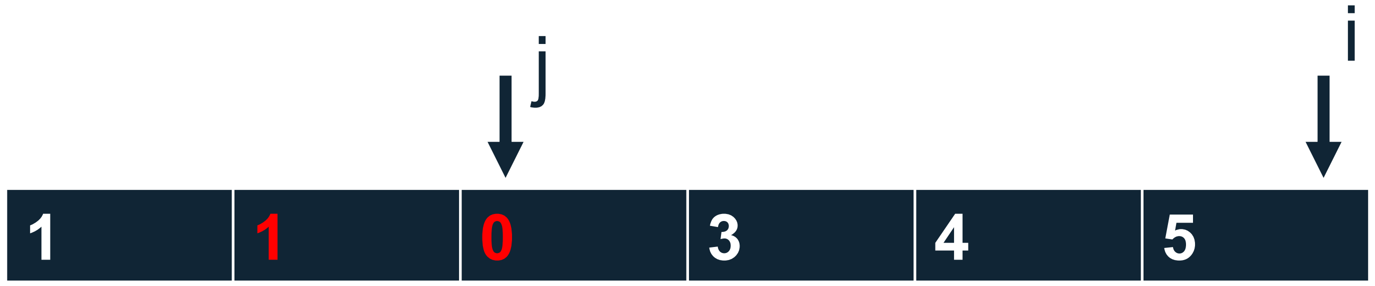
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



1	1	3	0	4	5

Insertion sort

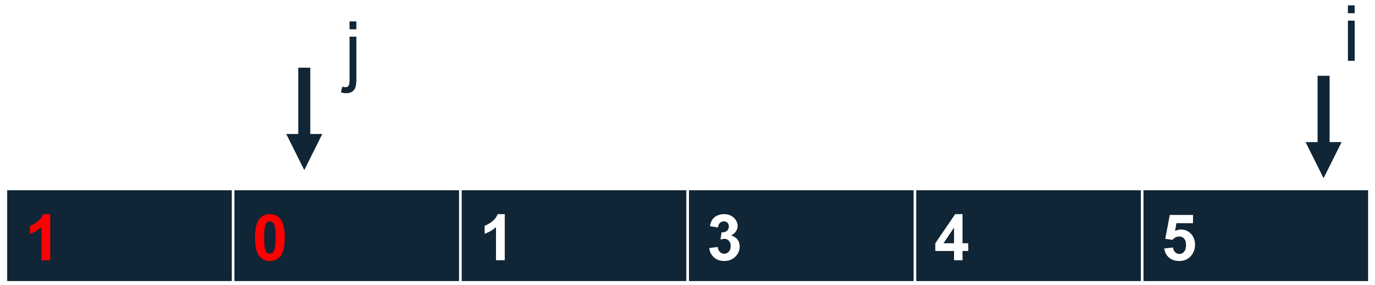
```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



1	1	0	3	4	5

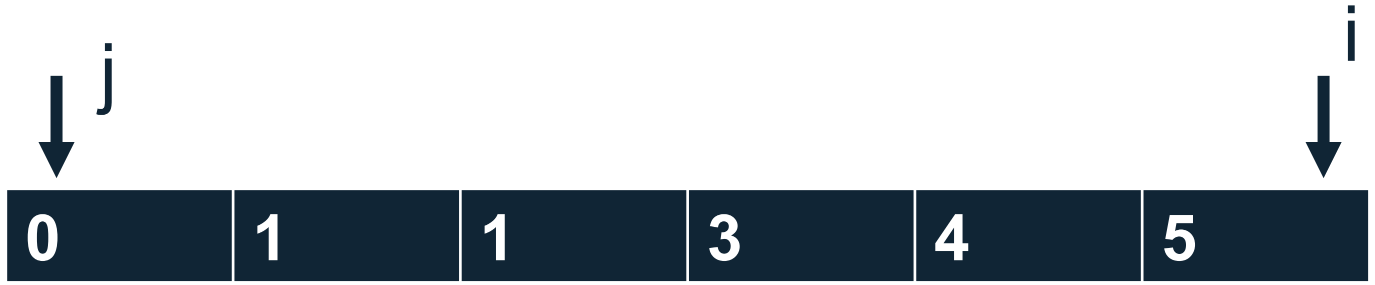
Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
               if(array.at(j) < array.at(j-1)) {
                    // swap array.at(j+1) and array.at(j)
               }
               else break;
          }
     }
}
```



Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
              if(array.at(j) < array.at(j-1)) {
                   // swap array.at(j+1) and array.at(j)
              }
              else break;
          }
     }
}
```



0	1	1	3	4	5

Insertion sort

```
void insertionSort(vector<int>& array) {
      for (int i = 1; i < array.size(); i++) {
          for(int j = i; j > 0; j--) {
               if(array.at(j) < array.at(j-1)) {
                    // swap array.at(j+1) and array.at(j)
               }
               else break;
          }
     }
}
```

Time complexity: O(n^2) in worst and average cases and O(n) in the best case.