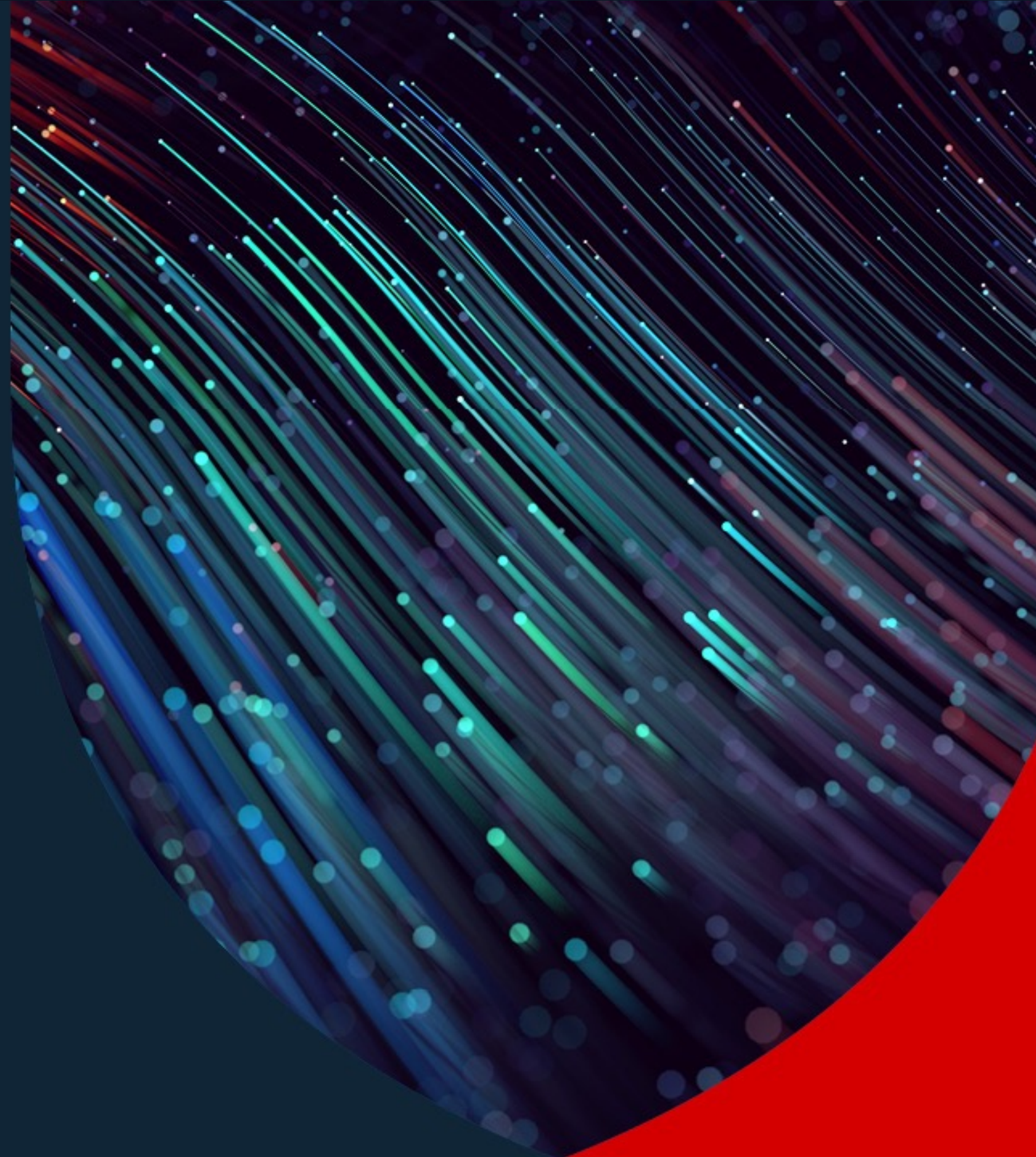
make history.

THE UNIVERSITY of ADELAIDE



Quick sort

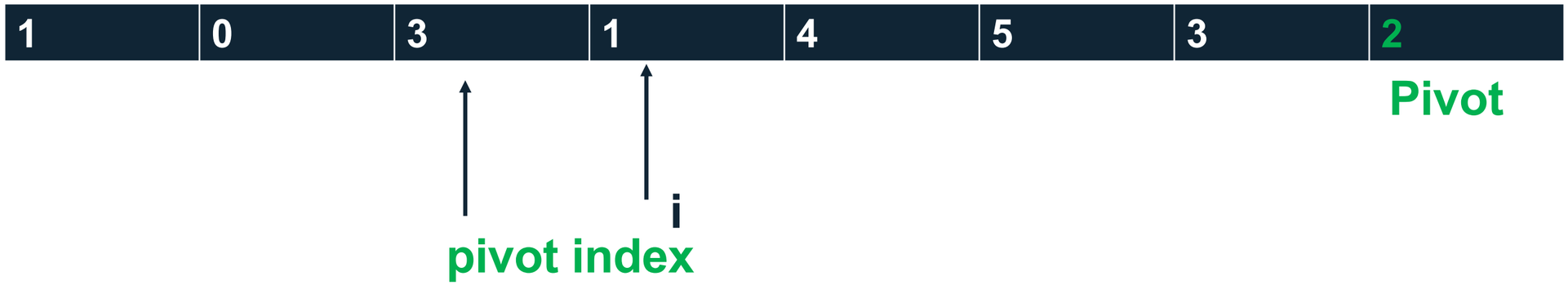
Dr. Anna Kalenkova

Quick sort

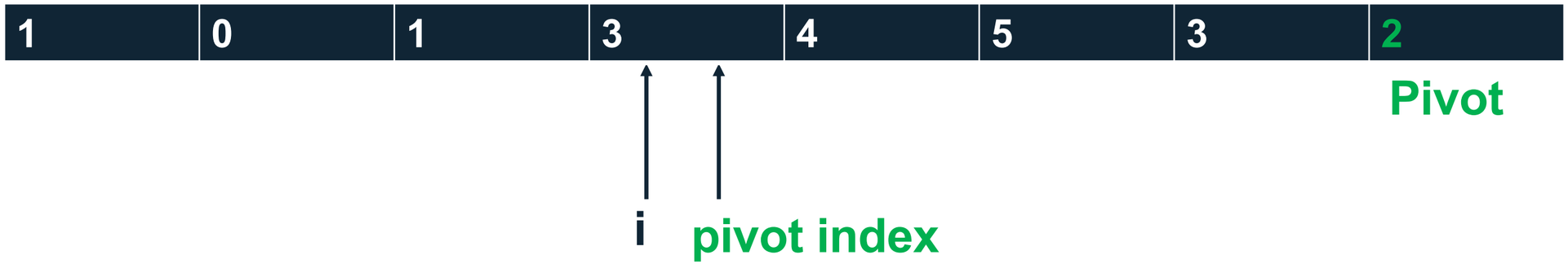
Quick sort

Quick sort

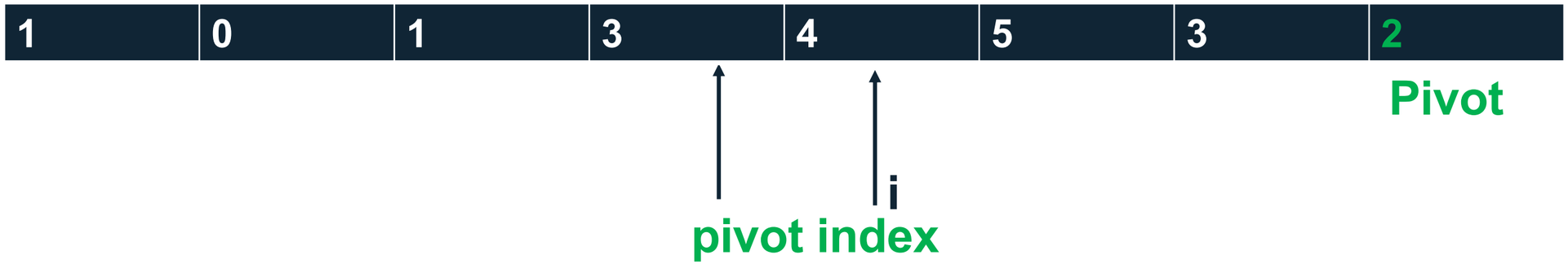
Quick sort



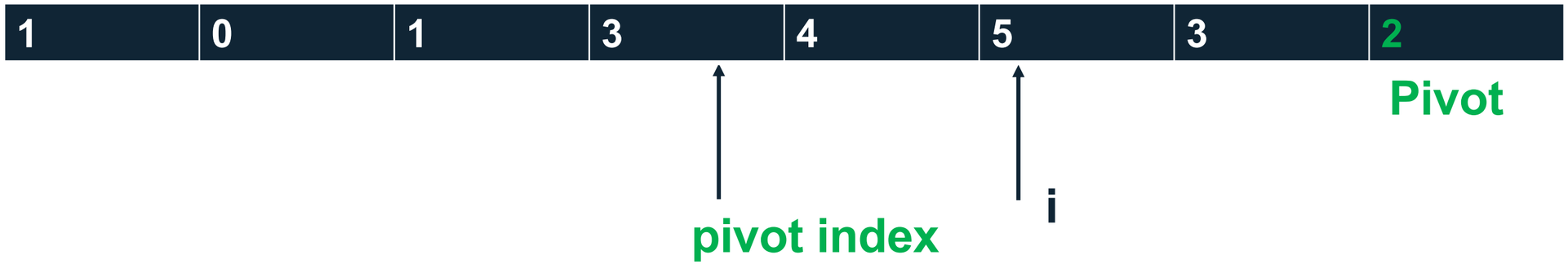
Quick sort



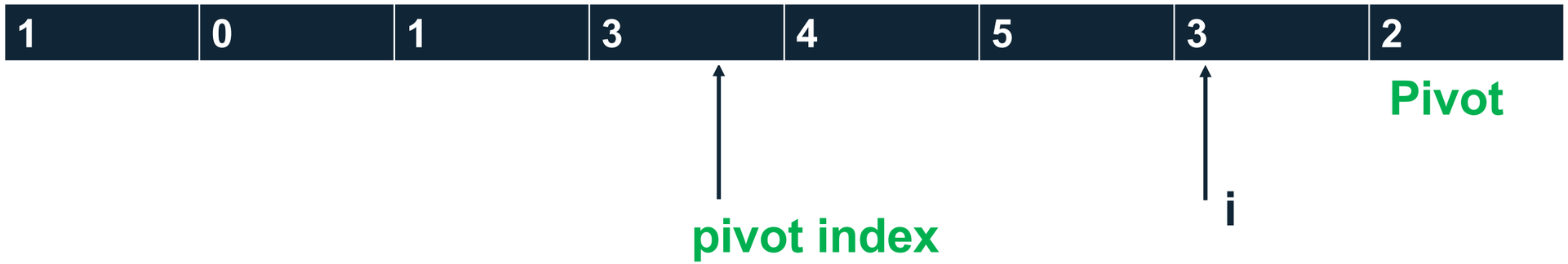
Quick sort



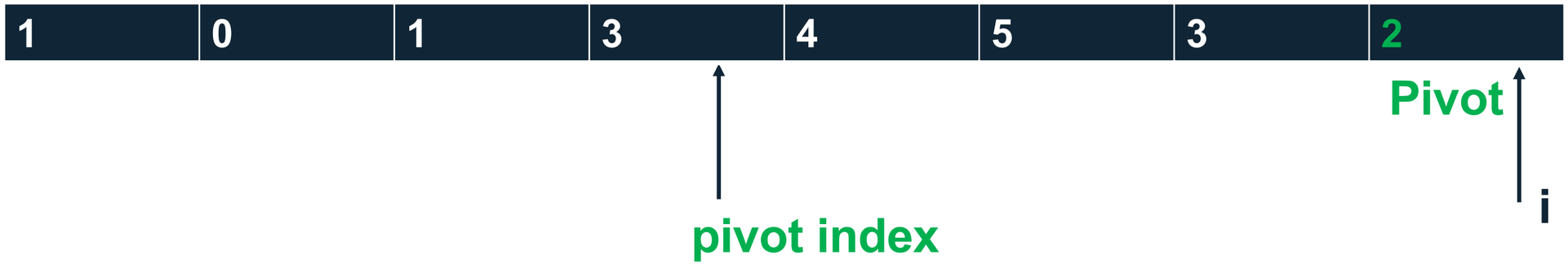
Quick sort



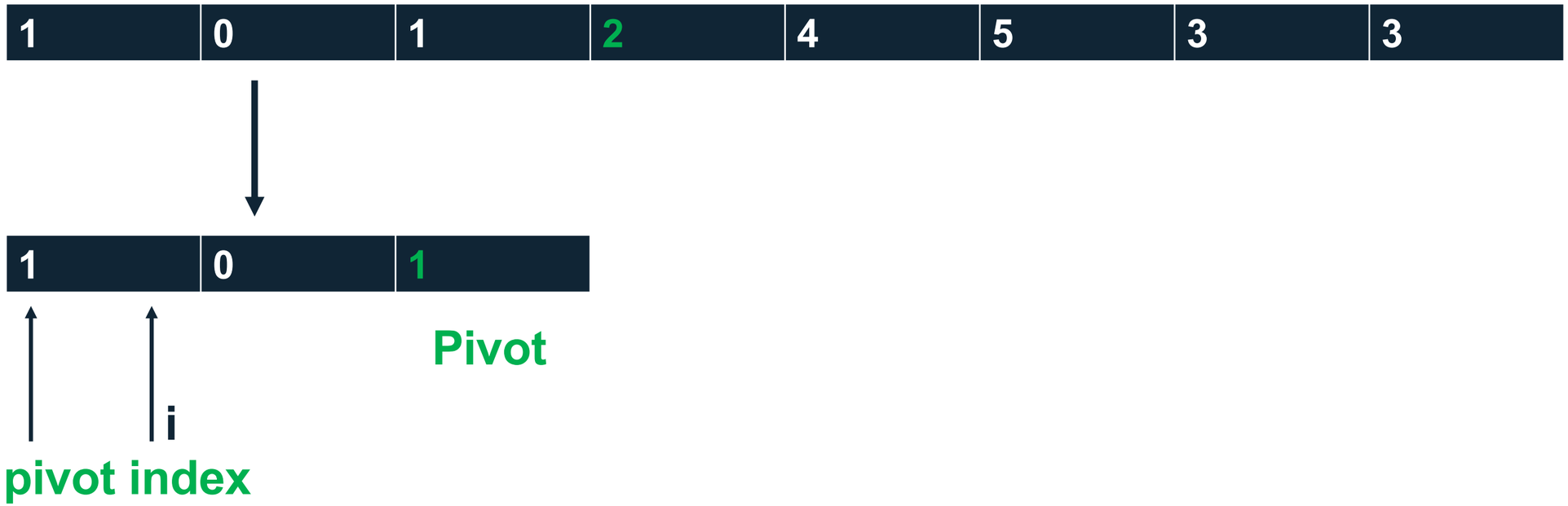
Quick sort



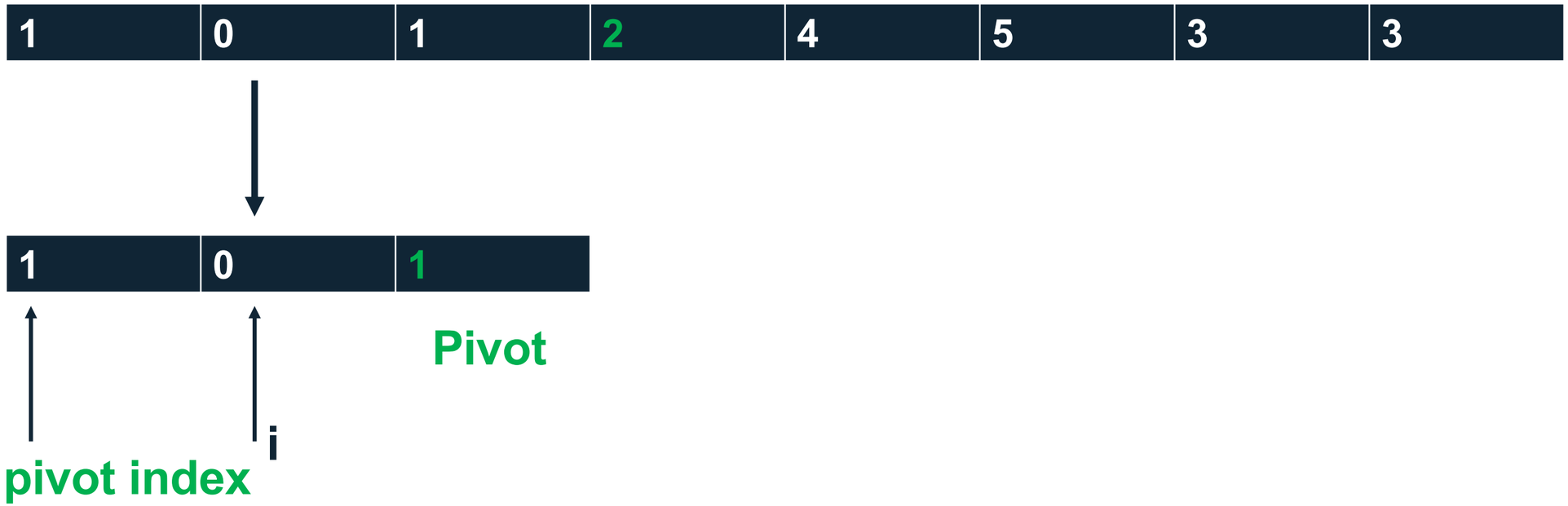
Quick sort



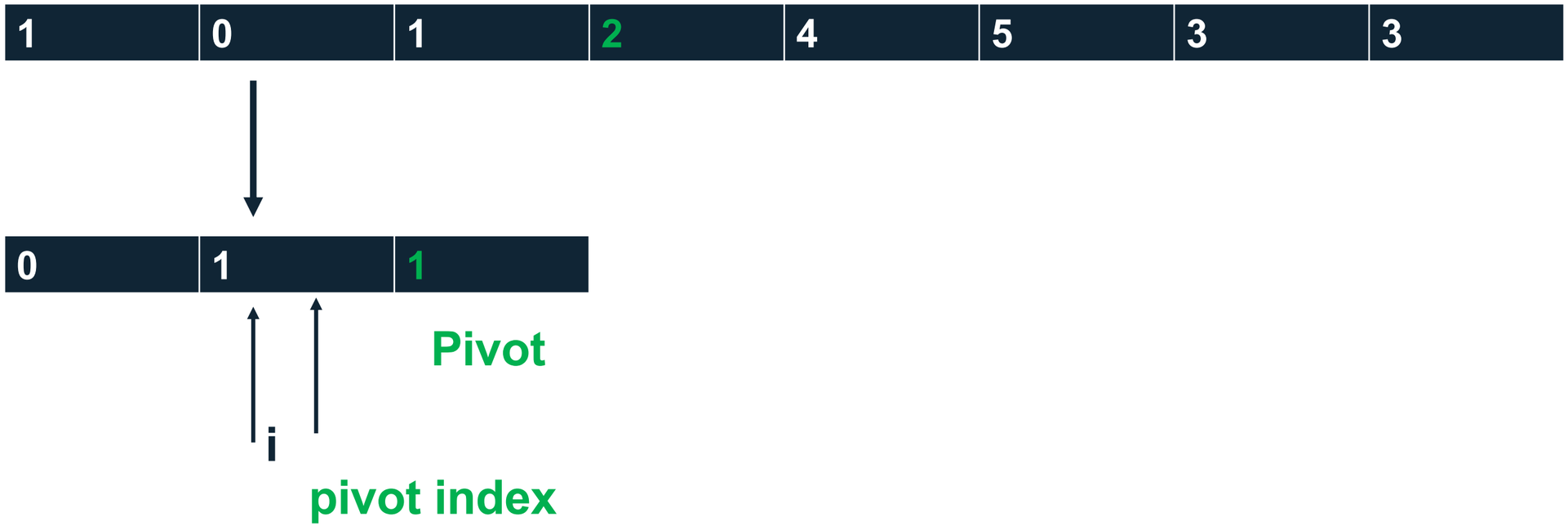
Quick sort



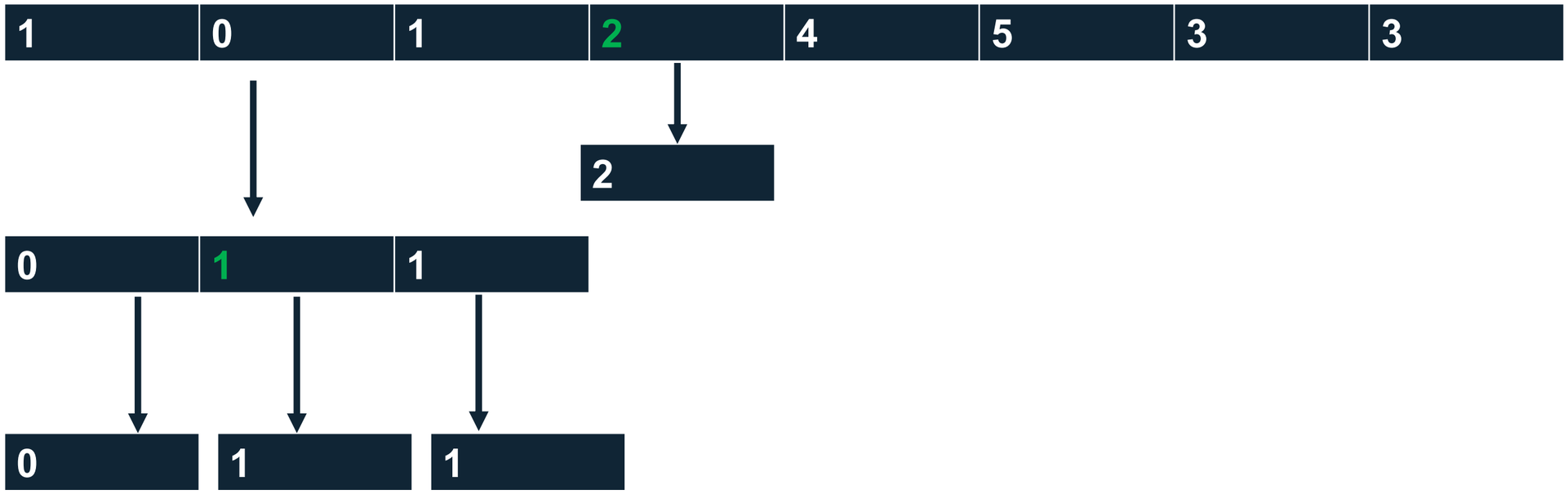
Quick sort



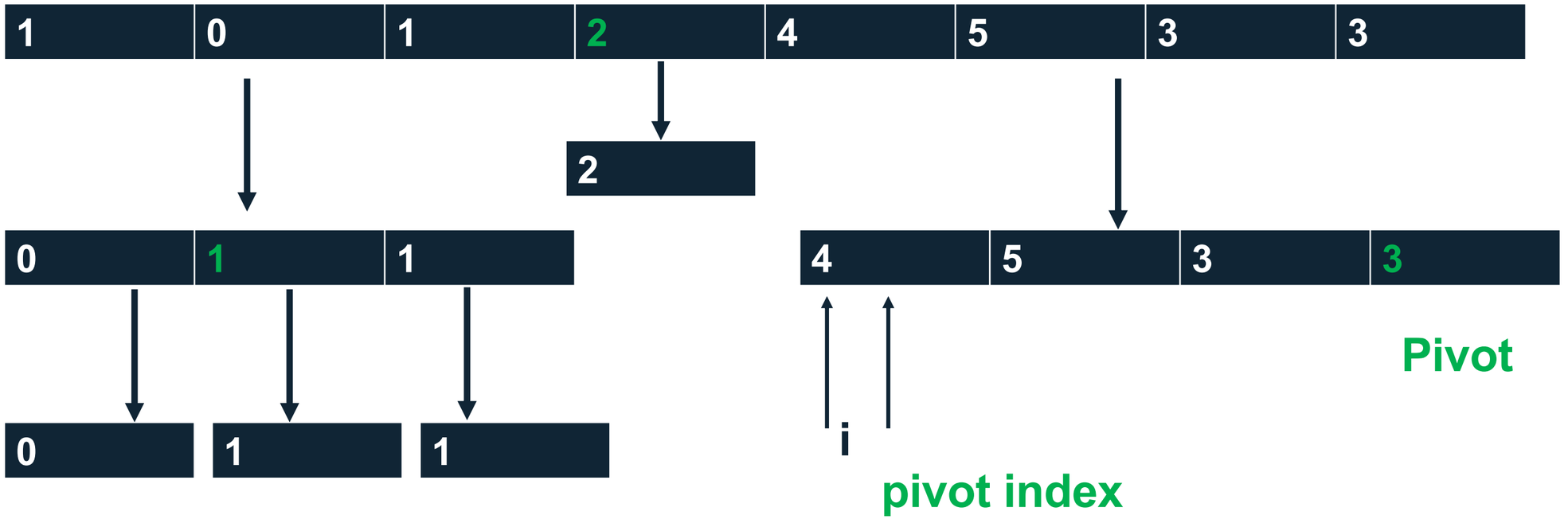
Quick sort



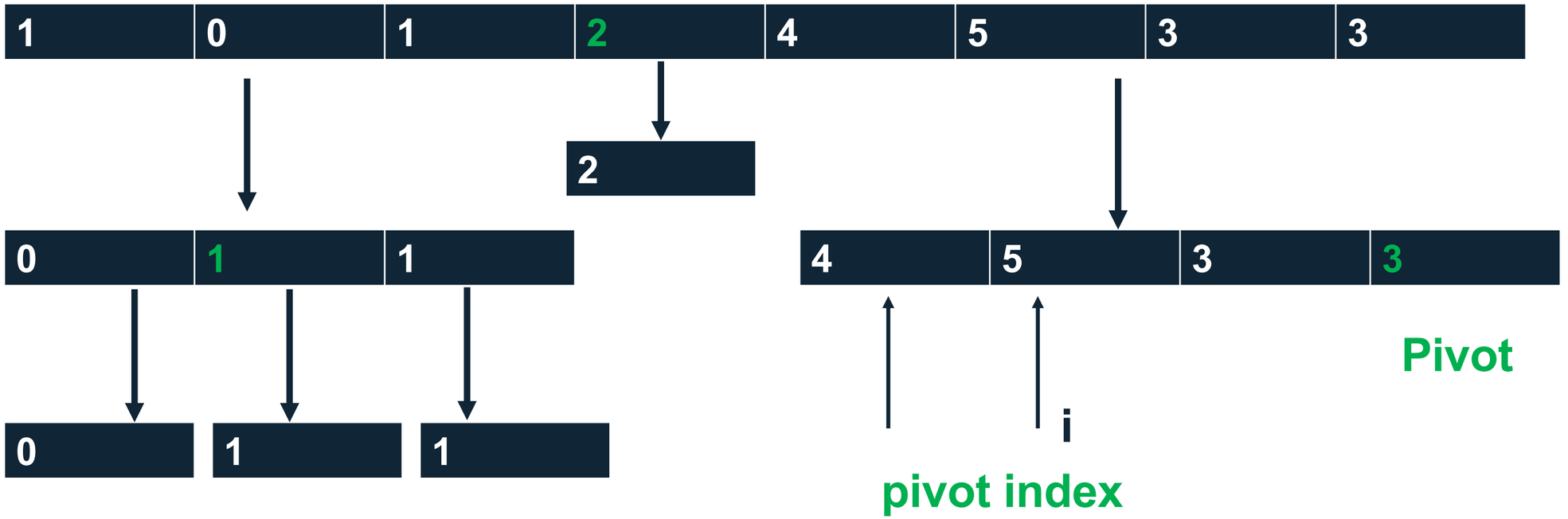
Quick sort



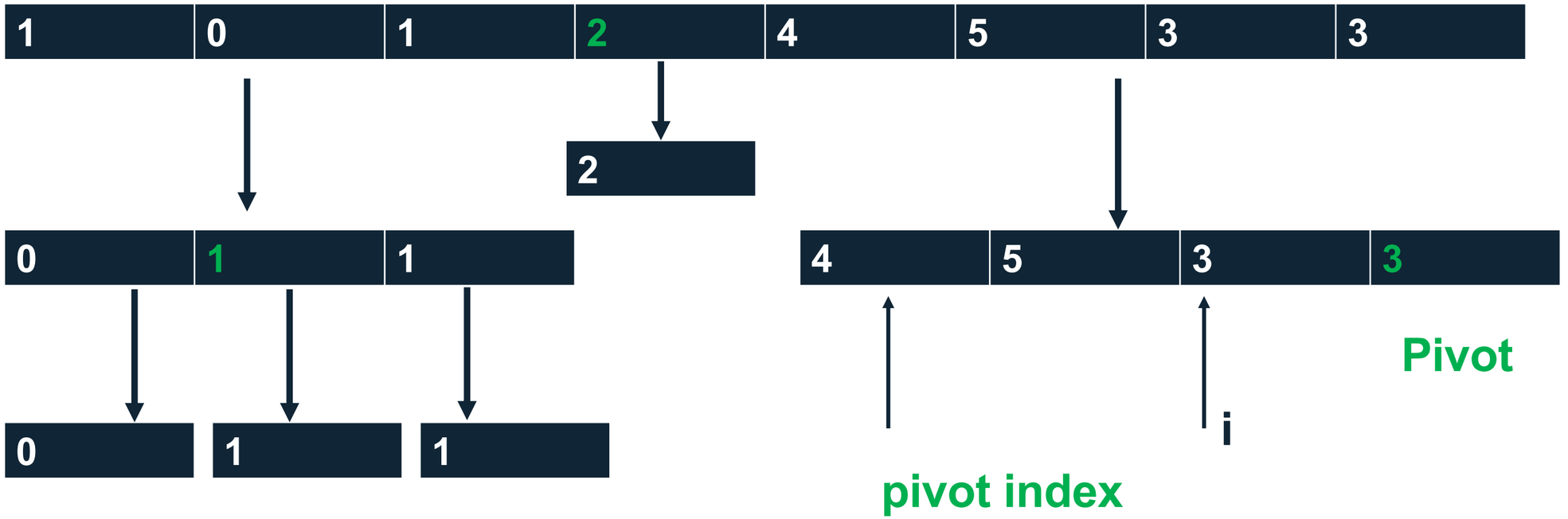
Quick sort



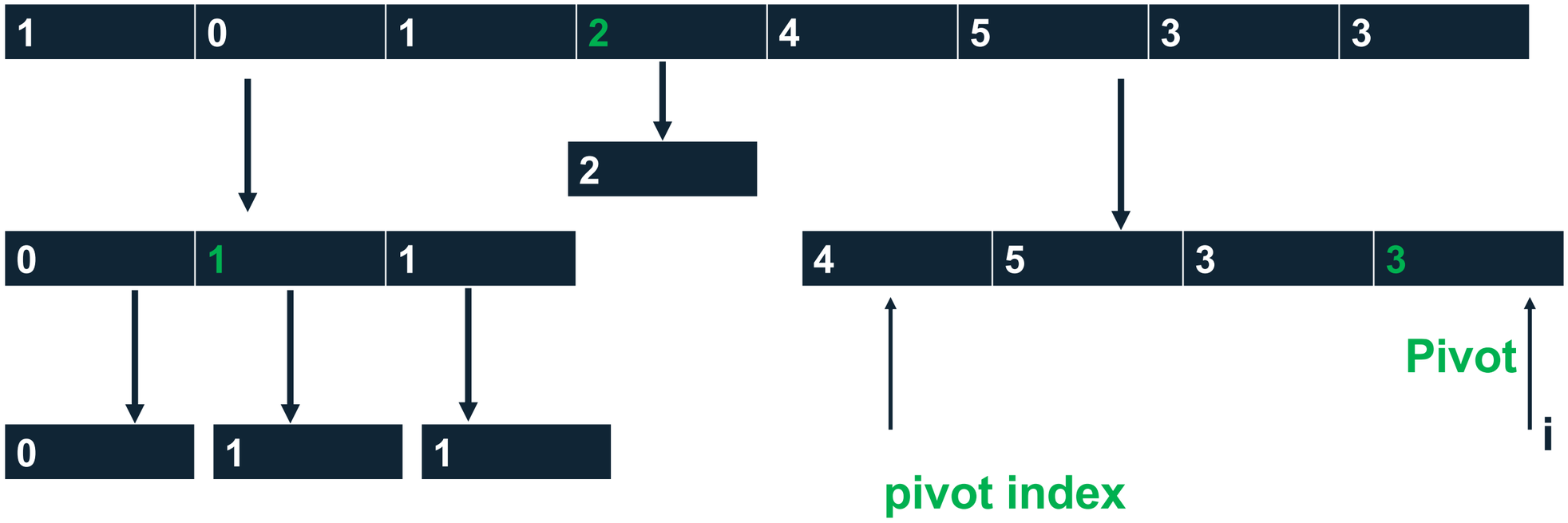
Quick sort



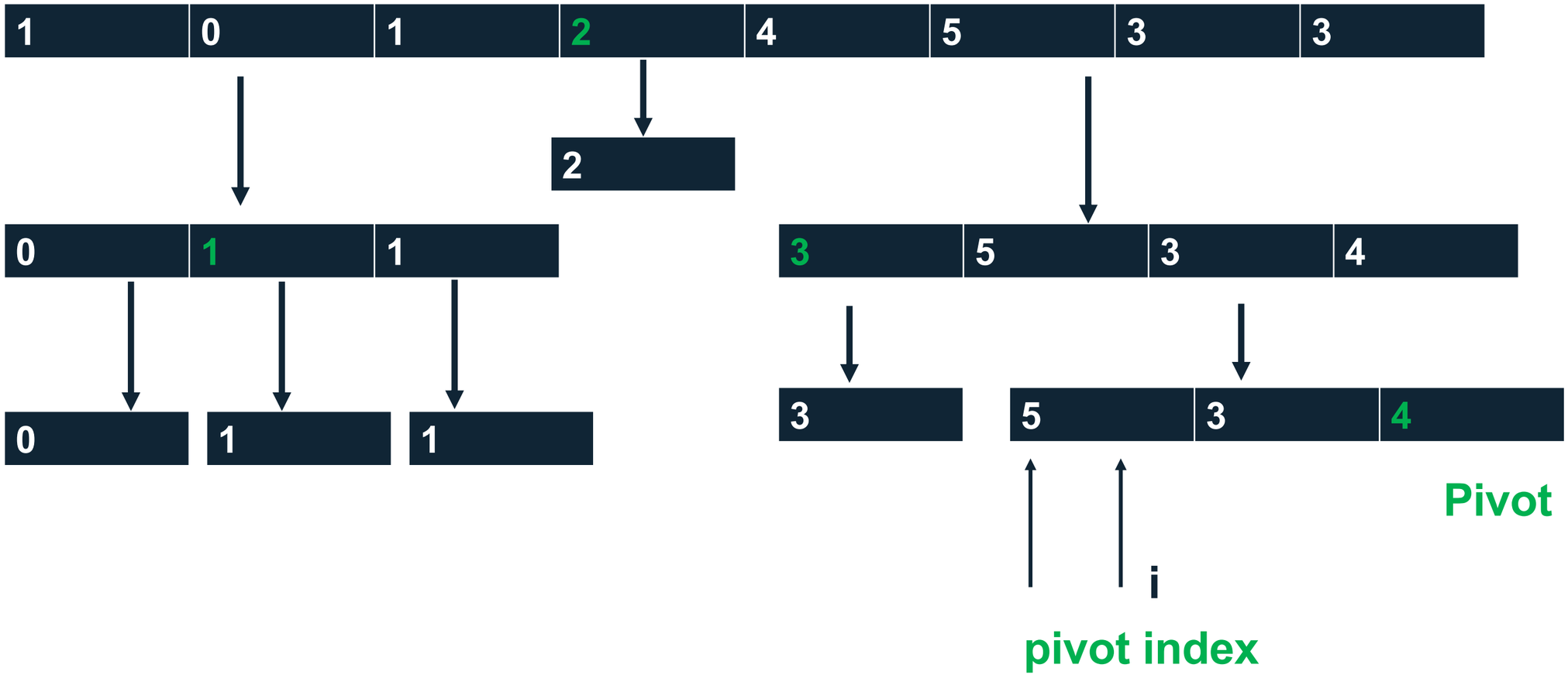
Quick sort



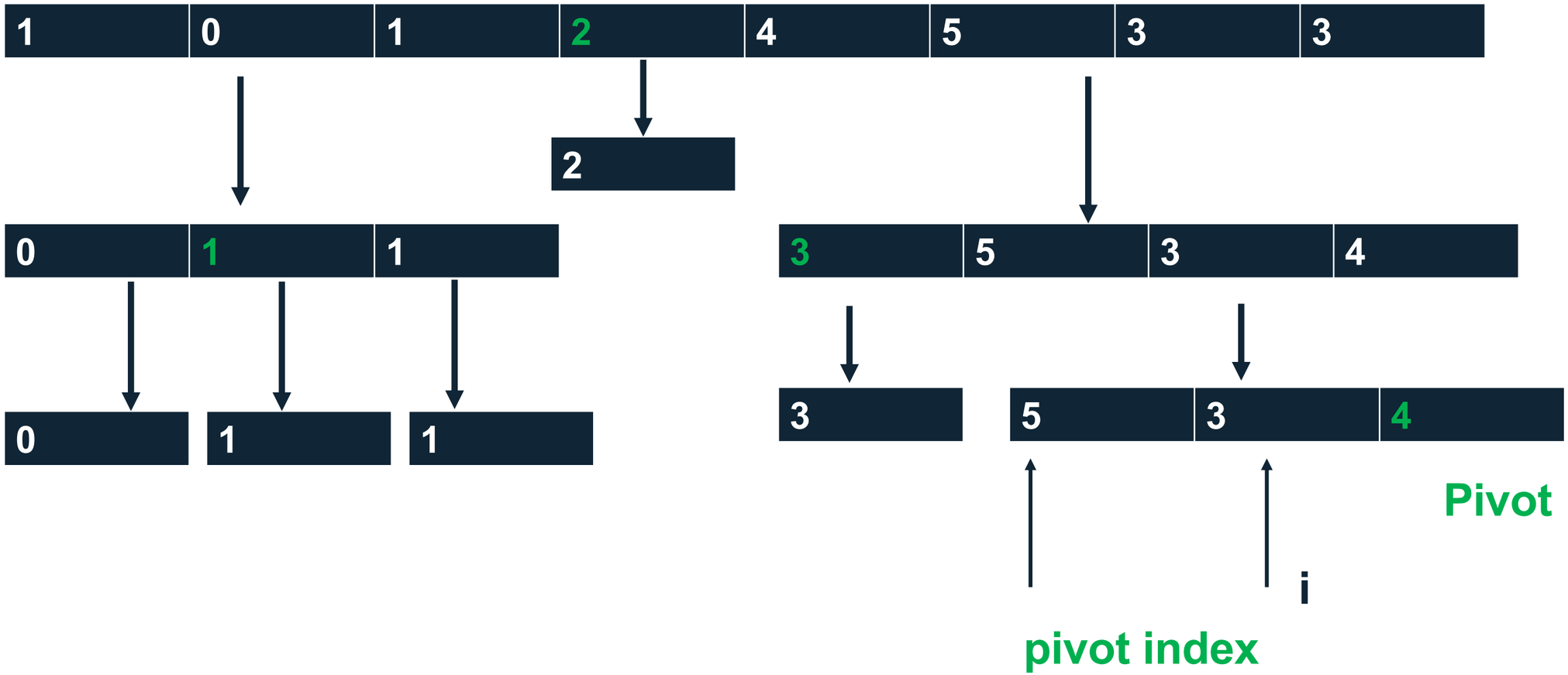
Quick sort



Quick sort

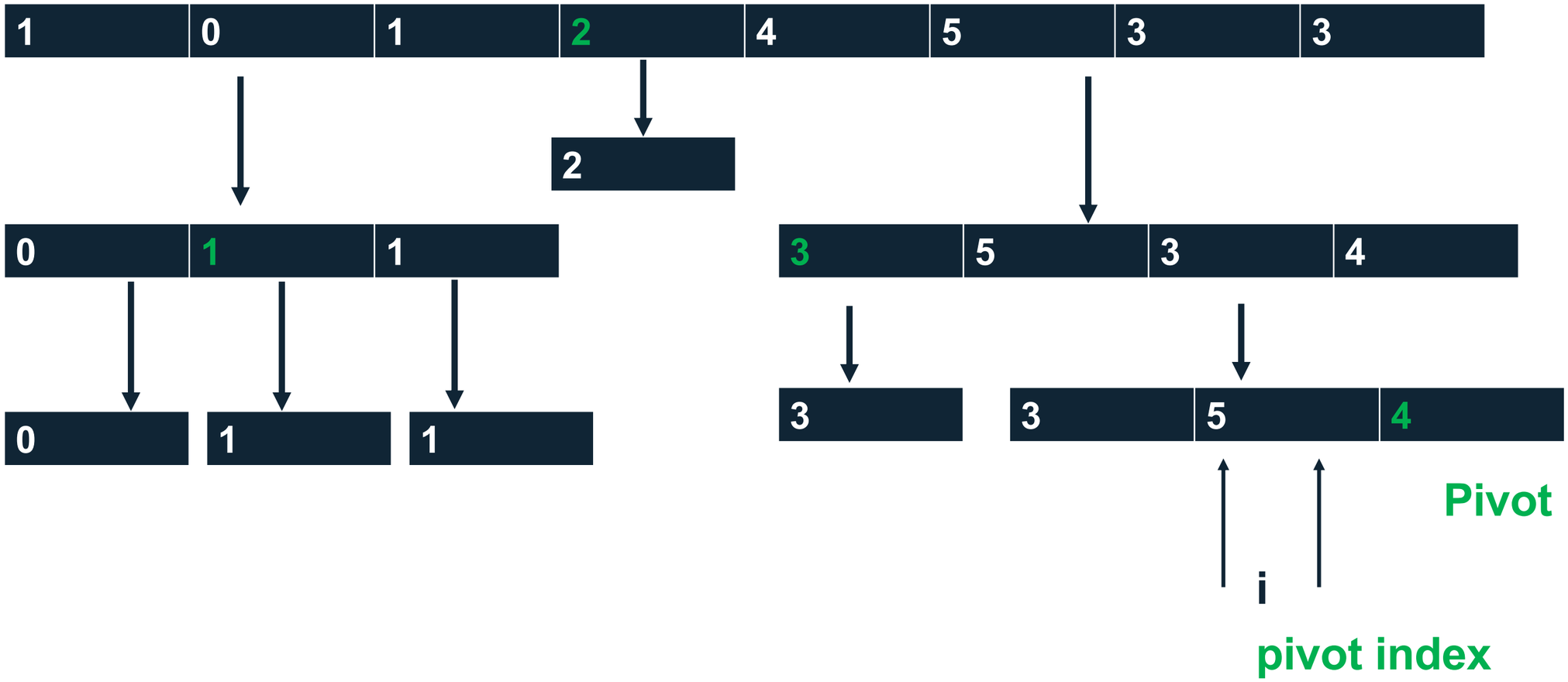


Quick sort



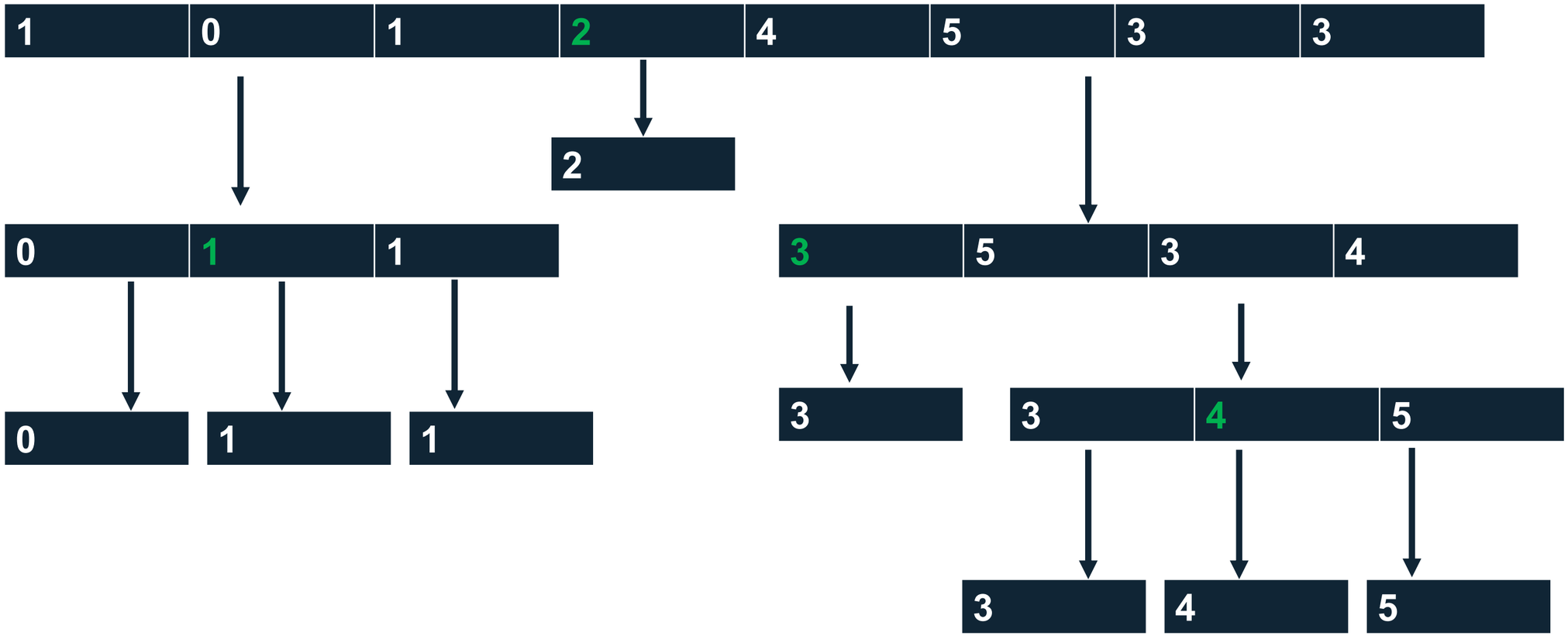
THE UNIVERSITY of ADELAIDE

Quick sort



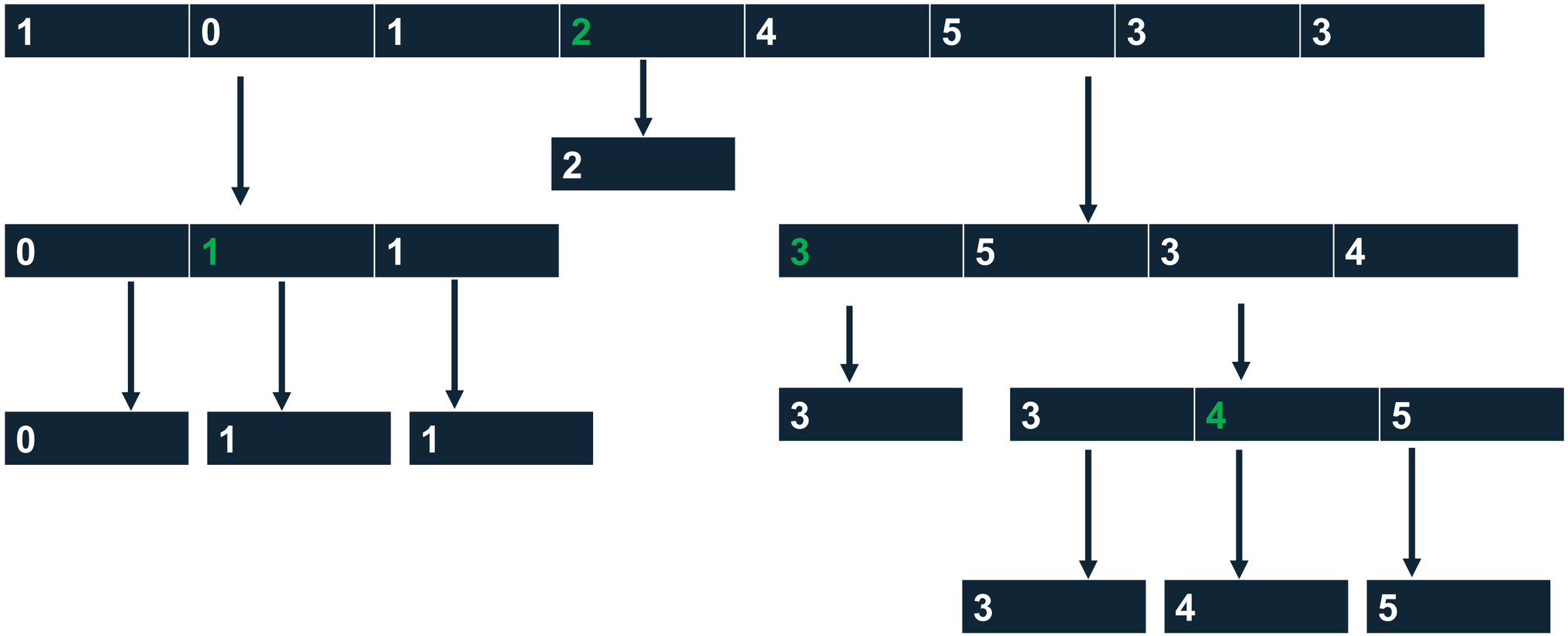
THE UNIVERSITY of ADELAIDE

Quick sort



THE UNIVERSITY of ADELAIDE

Quick sort



THE UNIVERSITY of ADELAIDE

Quick sort

```
void QuickSort::sort(vector<int>& array, int start, int end) {
     // Base case, nothing to sort
     if(start >= end) {
          return;
     }
     // Select the last element as pivot
     int pivot = array.at(end);

     int pivot_index = start;
     for(int i = start; i < end; i++) {
          if (array.at(i) < pivot) {
               // swap ith element and element at pivot_index
               …
               pivot_index++;
          }
     }
     // swap the pivot_index element and pivot
     …

     // call sort for subarrays
     sort(array, start, pivot_index - 1);
     sort(array, pivot_index + 1, end);
}
```

Quick sort

- The choice of pivot is essential.
- $O(n^2)$ is the complexity in the worst case.
- $O(n \log(n))$ is the complexity in best and average cases.