Selection sort

```
def selection_sort(A):
    n = len(A)
    for i in range(n):
        min_index = i
        for j in range(i + 1, n):
            if A[j] < A[min_index]:
                min_index = j
        A[i], A[min_index] = A[min_index], A[i]
```

- Runs in $O(n^2)$ time - 2 nested for-loops
- Repeatedly places min element at correct index
- Slow but intuitive, and doesn't require an extra output array

Merge sort

```
def merge_sort(A):
    if len(A) <= 1:
        return A

    mid = len(arr) // 2

    left = merge_sort(A[:mid])
    right = merge_sort(A[mid:])
    return merge(left, right)
```

- T(n) = T(n/2) + T(n/2) + T(merge)
- Divide and conquer approach
- Merge: merge two sorted arrays