# 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

# Merge sort (cont.)

```
def merge(left, right):
    merged = []
    l = 0
    r = 0
    while l < len(left) and r < len(right):
        if left[l] <= right[r]:
            merged.append(left[l])
            l += 1
        else:
            merged.append(right[r])
            r += 1
    merged.extend(left[l:])
    merged.extend(right[r:])
    return merged
```

- Merge the two sorted arrays using two pointers
- When first array is done, fill with second until it is also done
- Take smaller item until done
- Runs in O(n) time