Recurrence example: Merge sort

- Running times of recursive programs are solved using recurrence equations:

```
MergeSort(A, p, r)
// Let n = r − p + 1
if p < r then
        q ← (p + r)/2
        MergeSort(A, p, q)
        MergeSort(A, q + 1, r)
        Merge(A, p, q, r)
```

- Let $T(n)$ be the time taken by MergeSort to sort $n = r - p + 1$ elements.
- MergeSort recurrence: $T(n) = 2T(n/2) + cn$, $T(1) = 1$.

Merging two sorted subarrays

Algorithm to merge sorted subarrays $A[p..q]$ and $A[q+1..r]$. Its running time is $O(n)$, where $n = r - p + 1$.

```
Merge(A, p, q, r)
Copy A[p..q] into L[1..q − p + 1]
Copy A[q + 1..r] into R[1..r − q]
Set sentinels L[q − p + 2] = R[r − q + 1] = ∞
i ← 1;    j ← 1
for k ← p to r do
        if L[i] ≤ R[j] then
                A[k] ← L[i];    i ← i + 1
        else
                A[k] ← R[j];    j ← j + 1
```