Dr. A. Turing

Autumn Term 2026
Week 3

Sheet 3: Sorting and Recurrences: Suggested Solutions

Hand in your written solutions before next Tuesday's lecture.

Exercise 1 ***10 points***

Trace merge sort on the array $[5, 3, 8, 4, 2, 7, 1, 6]$, showing the array at each merge step.

Solution:
Splitting recursively gives $[5, 3, 8, 4]$ and $[2, 7, 1, 6]$, each split again down to single elements, then merged back up: $[3, 5]$, $[4, 8]$, $[1, 7]$, $[2, 6]$, then $[3, 4, 5, 8]$, $[1, 2, 6, 7]$, and finally $[1, 2, 3, 4, 5, 6, 7, 8]$.

Exercise 2 ***15 points***

Write down the recurrence relation for the running time of merge sort on an array of size n, and solve it using the recursion-tree method.

Solution:
The recurrence is $T(n) = 2T(n/2) + O(n)$. The recursion tree has $\log_2 n$ levels, each doing $O(n)$ total work across its nodes, so $T(n) = O(n \log n)$.

Exercise 3 ***20 points***

(a) Give a recurrence for the number of comparisons quicksort makes in the worst case (always picking the first element as pivot on an already-sorted array).

(b) Solve that recurrence and give its asymptotic complexity.

Solution:
(a) $T(n) = T(n-1) + (n-1)$, since the pivot never splits the array evenly and every other element is compared against it.

(b) Unrolling, $T(n) = \sum_{k=1}^{n-1} k = \binom{n}{2} = O(n^2)$.