

Wack.

This is a **regular task**. You must submit a PDF, which can be produced using the L^AT_EX template on Moodle, exported from a word processor, hand-written or any other method.

Consider the following algorithm:

WACKYSORT

- We are given an array A of length n .
- If $n \leq 3$, then we can sort A using selection sort.
- Otherwise:
 - First, recursively WACKYSORT the first $\lceil 3n/4 \rceil$ elements of the array.
 - Then, recursively WACKYSORT the *last* $\lceil 3n/4 \rceil$ elements.
 - Finally, recursively WACKYSORT the first $\lceil 3n/4 \rceil$ elements again.

In this task, we will analyse the WACKYSORT algorithm.

- (a) Prove that WACKYSORT correctly sorts an array.
- (b) Determine and analyse the time complexity of WACKYSORT.

Advice.

- (a) Prove that regardless of input (and its sortedness) that the output contains the same elements as the input, and that the output is sorted.

Your proof should be clearly structured, should flow well, not omit major steps; you should also conclude appropriately.

- (b) You should clearly state the correct running time. You should justify (using the Master Theorem, or otherwise) your running time.

Expected length: Up to a page in total.