Exercise Sheet 3
COMS10017 Algorithms 2020/2021

Reminder: $\log n$ denotes the binary logarithm, i.e., $\log n = \log_2 n$.

1 Warm up: Proof by Induction

Consider the following sequence: $s_1 = 1, s_2 = 2, s_3 = 3$, and $s_n = s_{n-1} + s_{n-2} + s_{n-3}$, for every $n \geq 4$. Prove that the following holds:

$$s_n \leq 2^n \ .$$

2 Loop Invariant

Prove that the stated invariant holds throughout the execution of the loop (using the Initialization, Maintenance, Termination approach discussed in the lectures):

Algorithm 1 Algorithm $\mathcal{A}$

Require: Array A of length n ($n \geq 2$)

```
1: S ← A[0] − A[1]
2: for i ← 1 . . . n − 2 do
3:     S ← S + A[i] − A[i + 1]
4: end for
5: return S
```

Invariant:

At the beginning of iteration i, $S = A[0] - A[i]$ holds.

What does the algorithm compute?

3 Insertionsort

What is the runtime (in Θ-notation) of Insertionsort when executed on the following arrays of lengths n:

1. $1, 2, 3, 4, \ldots, n-1, n$
2. $n, n-1, n-2, \ldots, 2, 1$
3. The array A such that $A[i] = 1$ if $i \in \{1, 2, 4, 8, 16, \ldots\}$ (i.e., when i is a power of two) and $A[i] = i$ otherwise.

4 Mergesort

The Mergesort algorithm uses the MERGE operation, which assumes that the left and the right halves of an array A are already sorted, and merges these two halves so that A is sorted afterwards. The runtime of this operation is $O(n)$.

Suppose that we replaced the MERGE operation in our Mergesort algorithm with a less efficient implementation that runs in time $O(n^2)$ (instead of $O(n)$). What is the runtime of our modified Mergesort algorithm?

5 Runtime Analysis

Algorithm 2

Require: Integer $n \geq 2$
  $x \leftarrow 0$
  $i \leftarrow n$
  while $i \geq 2$ **do**
    $j \leftarrow \lceil n^{1/4} \rceil \cdot i$
    while $j \geq i$ **do**
      $x \leftarrow x + 1$
      $j \leftarrow j - 10$
    end while
    $i \leftarrow \lfloor i/\sqrt{n} \rfloor$
  end while
  return x

Determine the runtime of Algorithm 3 in Θ-notation.