Homework Sheet 1

Author: Abdullah Oğuz Topçuoğlu

Exercise 1

(a)

The algorithm is correct because when we sort the numbers, if there is no missing number, then each number has an index one more than its value. (thats the case because the array given is 1-indexed but the values we are given start from 0) And the algorithm starts from the first element and when it encounters the first number that is equal to its index (rather than being equal to index + 1) it means we just skipped a number and we return it. And if we havent encounter any number in process then it means it has to be the last element in the sorted array which has the value "n" always (because it is sorted)

(b)

So we have an accumulator "y" and in each iteration we add the current index and we substract the current value in the array. So the value of "y" at the end is:

$$\begin{aligned}\sum_{i=1}^{n} i - A[i] &= \sum_{i=1}^{n} i - \sum_{i=1}^{n} A[i] \\ &= \frac{n(n+1)}{2} - (0 + 1 + 2 + ... + n - x) \quad \text{(where x is the missing number)} \\ &= \frac{n(n+1)}{2} - \left(\frac{n(n+1)}{2} - x\right) \\ &= x\end{aligned}$$

So "y" is equal to the missing number at the end of the algorithm. And we return "y" at the end.

Exercise 2

We have an accumulator "y" which is initialized to 0. And in each iteration we update "y" to be equal to the current coefficient + z times the previous value of "y". So if we unroll the loop, we can see that at the end of the algorithm, "y" is equal to:

$$\begin{aligned} y &= a_0 + z \cdot (a_1 + z \cdot (a_2 + z \cdot (... + z \cdot (a_{n-1} + z \cdot (a_n + z \cdot 0))...))) \\ &= a_0 + a_1 z + a_2 z^2 + ... + a_{n-1} z^{n-1} + a_n z^n \\ &= \sum_{i=0}^{n} a_i z^i \\ &= p(z) \end{aligned}$$

Exercise 3

(a)

There exists constants c_1 and c_2 such that:

$$\begin{aligned} f_1(n) &\leq_{ae} c_1 g_1(n) \\ f_2(n) &\leq_{ae} c_2 g_2(n) \end{aligned}$$

Let $c = \max(c_1, c_2)$, then:

$$\begin{aligned} f_1(n) + f_2(n) &\leq_{ae} c_1 g_1(n) + c_2 g_2(n) \\ &\leq_{ae} c g_1(n) + c g_2(n) \\ &\leq_{ae} c(g_1(n) + g_2(n)) \\ &\leq_{ae} c \cdot 2 \max(g_1(n), g_2(n)) \end{aligned}$$

Thats what we wanted to show.

(b)

There exists constants c_1 and c_2 such that:

$$\begin{aligned} f_1(n) &\leq_{ae} c_1 g_1(n) \\ f_2(n) &\leq_{ae} c_2 g_2(n) \end{aligned}$$

Then:

$$\begin{aligned} f_1(n) \cdot f_2(n) &\leq_{ae} c_1 g_1(n) \cdot c_2 g_2(n) \\ &= (c_1 c_2)(g_1(n) \cdot g_2(n)) \end{aligned}$$

Thats what we wanted to show.

(c)

There exists constants c_1 and c_2 such that:

$$\begin{aligned} f(n) &\leq_{ae} c_1 g(n) \\ g(n) &\leq_{ae} c_2 h(n) \end{aligned}$$

Then:

$$\begin{aligned} f(n) &\leq_{ae} c_1 g(n) \\ &\leq_{ae} c_1 c_2 h(n) \end{aligned}$$

Thats what we wanted to show.

Exercise 4

The order is:

$$\log n < n \log\log n < n \log n < 2n^2 - 4n < n^2 \log n < 2^n$$

$\log n < n \log\log n$:
From exercise 3(b), if we set $f_1(n) = \log n$, $g_1(n) = n$, $f_2(n) = 1$, $g_2(n) = \log\log n$, we have:

$$\begin{aligned} f_1(n) \cdot f_2(n) &\leq_{ae} c_1 g_1(n) \cdot c_2 g_2(n) \\ \log n &\leq_{ae} cn \log\log n \end{aligned}$$

$n \log n < 2n^2 - 4n$:
We can see that for $n \geq 4$:

$$\begin{aligned} 2n^2 - 4n - n\log n &= n(2n - 4 - \log n) \\ &\geq n(2n - 4 - n) \\ &= n(n-4) \\ &\geq 0 \end{aligned}$$

$2n^2 - 4n < n^2 \log n$:
We can see that for $n \geq 4$:

$$\begin{aligned} n^2 \log n - 2n^2 + 4n &= n(n \log n - 2n + 4) \\ &\geq n(n \log n - 2n + 2n) \\ &= n^2 \log n \\ &\geq 0 \end{aligned}$$

$n^2 \log n < 2^n$:
We can see that for $n \geq 16$:

$$\begin{aligned} 2^n - n^2 \log n &\geq 2^n - n^2 \cdot n \\ &= 2^n - n^3 \\ &\geq 0 \end{aligned}$$