

# Algorithms example sheet 1

---

## Exercise 1

Assume that each `swap(x, y)` means three assignments (namely `tmp = x; x = y; y = tmp`). Improve the `insertSort` algorithm pseudocode shown in the hand-out to reduce the number of assignments performed in the inner loop

```
def insertSort(a):
    for i from 1 included to len(a) excluded:
        j = i - 1
        current = a[i]
        while j >= 0 and a[j] > current:
            a[j+1] = a[j]
            j = j - 1
        a[j] = current
```

## Exercise 3

$$|\sin(n)| = O(1)$$

$$|\sin(n)| \neq \Theta(1)$$

$$200 + \sin(n) = \Theta(1)$$

$$123456n + 654321 = \Theta(n)$$

$$2n - 7 = O(17n^2)$$

$$\lg(n) = O(n)$$

$$\lg(n) \neq \Theta(n)$$

$$n^{100} = O(2^n)$$

$$1 + 100/n = \Theta(1)$$

For each of the above “=” lines, identify the constants  $k, k_1, k_2, N$  as appropriate. For each of the “ $\neq$ ” lines, show they can’t possibly exist.

1.  $k = 1, N = 0$

2. Since  $k_1 > 0$ , and  $|\sin(n)|$  falls to zero infinitely many times, there is no  $N$  such that  $|\sin(n)| > k_1$  for  $n > N$ .

3.  $k_1 = 199, k_2 = 201, N = 0$

4.  $k_1 = 123456, k_2 = 777777, N = 1$

5.  $k = 1, N = 0$

6.  $k = 1, N = 0$

7.  $k_1 n = \ln(n) \implies k_1 = \ln(n)/n$ , which has solutions for all  $0 < k_1 \leq 1/e$ , so no  $k_1 > 0$  is small enough to provide a lower bound.

8.  $k = 1, N = 1000$

9.  $k_1 = 1, k_2 = 101, N = 1$

## Exercise 8

*Prove that Bubble sort will never have to perform more than  $n$  passes of the outer loop.*

In each pass of the inner loop, if an item is larger than the next in the list, then they are swapped.

This means that during the outer loop, once the largest item is found, it is swapped with each successive item until it ends up at the end of the list. The same then happens with the second largest, then the third, etc. until the  $n$ th largest item is placed, i.e. the smallest item, in which case the algorithm is complete.