1 Iterative Algorithms

An **iterative algorithm** is encoded without loops and with procedure or function calls taking constant time.

If a loop is executed $O(f(n))$ times and each iteration takes $O(g(n))$ time, then the entire loop takes the time of $O(f(n)g(n))$.

If an if-then-else statement has corresponding complexities of $O(h(n))$, $O(f(n))$, and $O(g(n))$, then the complexity of the statement is $O(\max\{h(n), f(n), g(n)\})$.

For procedural or functional calls, if P, with the size of input m, has a running time $T(m) \in O(f(m))$,then a call to P with an input of size $g(n)$ takes the time of $O(f(g(n)))$.

Let $T_{IS} : \mathbb{N} \to \mathbb{N}$ be such that $T_{IS}(n)$ is the maximum number of comparisons and assignments taken by insertion sort on arrays A of length n.

An Insertion Sort algorithm on the list A is as follows:

(1) $i \leftarrow 1$

(2) while $i \leq \text{length}(A)$ do

(3)   while $j > 1$ and $A[j] < B[j-1]$ do

(4) $B[j] \leftarrow B[j-1]$

(5) $j \leftarrow j - 1$

(6) $B[j] \leftarrow A[i]$

(7) $i \leftarrow i + 1$

(8) return(B)

Lemma 1.1

$T_{IS}(n) \leq 2n^2 + 4n + 2$

Proof.

Let $n \in \mathbb{N}$ be arbitrary.

Let A be an arbitrary array of length n.

From the code, there are n complete iterations of the outer while loop.

Each iteration of the outer while loop consists of 4 steps (on lines 2, 3, 7, 8) plus an execution of the inner while loop.

For each complete iteration of the inner while loop at most 3 steps are performed (on lines $4, 5, 6$).

During the ith iteration of the outer loop, there are at most $i - 1$ complete iterations of the inner while loop.

The final (incomplete) iteration of the inner while loop takes at most 2 steps.

Therefore, the number of steps taken by the inner while loop is at most $4(i-1) + 2$.

Thus, the total number of steps taken by the n complete iterations of the outer while loop is at most

$$\sum_{i=1}^{n}[(4i-1)+2+4] = 2n^2+4n.$$

The final incomplete iteration of the outer while loop takes 1 step and there is 1 step before the outer while loop.

Hence $T_{IS}(n) \leq 2n^2+4n+2$ by generalisation. $\square$

Claim. For $i \in \{1, \cdots, n\}$, $4i+1$ steps are performed during the iteration i of the outer while loop on input A. After the iteration is completed, B contains the elements $\{n-i+1, \cdots, n\}$.

Proof.

Let $P(i)$ ="$4i+1$ steps are performed during the iteration i and afterwards $\{n-i+1, \cdots, n\} \subseteq B$".

Base Case: $i=1$.

5 steps are performed in iteration 1 (on lines 2. 3, 4, 7 and 8).

By line 7, B contains $\{n\}$. Hence, $P(1)$ holds.

Let $i < n$ and assume $P(i)$.

During the iteration $i+1$ of the outer while loop there are 4 steps in addition to the inner loop (on lines 2, 3, 7, 8).

There are i complete iterations of the inner while loop.

Since $A[i+1] = n-i$, which is smaller than all the elements in b, each iteration of the inner loop takes 4 steps (on lines 4, 5, 6). In the final incomplete iteration of the inner while loop, 1 step is performed and $j=1$, so $A[i+1] = n-i$ is written into $B[i]$, while all the other elements of B have been shifter right.

Thus, after the iteration $\{n, \cdots, n\} \subseteq B$ and $4i+5$ steps are performed during the iteration $i+1$. Thus, $P(i+1)$ is true.

The final iteration of the outer while loop takes 1 step on line 4, while there is also 1 step taken before the outer loop on line 1.

Thus $t_{IS}(A) = 2 + \sum_{i=1}^{n}(4i+1)$, so $T_{IS}(n) \geq 2 + \sum_{i=1}^{n}(4i+1)$, and thus $T_{IS}(n) \leq 2n^2+3n+2$.

$\square$