Tutorial 1

Example 1

$x := 1$
repeat
  $x := x * 2$
until (NOT $x < 20$)
$x := x + 1$

Similarly, the same can be achieved also by using a **while**-loop.

Example 2

INPUT: A sequence S of natural numbers of length n such that $n > 0$.
OUTPUT: The smallest and largest element of S.

- $min := S[1]$; $max := S[1]$
  for $i := 2$ to n **do**
    if $S[i] < min$ **then**
      $min := S[i]$
    else
      if $S[i] > max$ **then**
        $max := S[i]$
      end if
    end if
  end for
  return min, max

- In the worst-case the number of comparisons is $2(n-1) = 2n - 2$.

- Yes. For example let $n = 2$ and let $S = [3, 4]$ (i.e. $S[1] = 3$ and $S[2] = 4$). Then the algorithm performs 2 comparisons. If $S = [6, 5]$ (i.e. $S[1] = 6$ and $S[2] = 5$) the algorithm performs only 1 comparison.
  Remark: For your algorithm the answer can be different.

Example 3

INPUT: A sequence S of natural numbers of length n such that $n > 0$.
OUTPUT: The second smallest ($ssmall$) element of S.

- $min := \infty$; $ssmall := \infty$
  for $i := 1$ to n **do**
    if $S[i] < ssmall$ **then**
      if $S[i] < min$ **then**
        $ssmall := min$

```
            min := S[i]
          else
            ssmall := S[i]
          end if
        end if
      end for
      if ssmall = ∞ then
        return "second smallest does not exist"
      else
        return ssmall
      end if
```

- Number of comparisons in the worst-case is $2n + 1$.
- Yes.
- For each n, the algorithm performs the largest number of comparisons on e.g. the following sequence: $S = [n, n-1, n-2, \dots, 2, 1]$ (i.e. $S[1] = n$, $S[2] = n-1$, $S[3] = n-2$, …, $S[n-1] = 2$, $S[n] = 1$).

Example 4

INPUT: Text x of length n and pattern y of length m such that $n > m > 0$.
OUTPUT: Number of occurrences of y in x.

- ```
  occurrences := 0
  for i := 1 to n − m + 1 do
    j := 1
    while x[i + j − 1] = y[j] and j ≤ m do
      j := j + 1
    end while
    if j = m + 1 then
      occurrences := occurrences + 1
    end if
  end for
  return occurrences
  ```
- Number of comparisons in the worst-case (we count only the comparisons between two positions in x and y): $(n - m + 1) \cdot m = nm - m^2 + m = O(nm)$ because $n > m$.
- For each n and m, the largest number of comparisons will be e.g. on the input $x = [\underbrace{A, A, \dots, A}_{n\times}]$ and $y = [\underbrace{A, A, \dots, A}_{m\times}]$.
  ```