# Time complexity

Week 13

# The running time

- Typically, the running time of an algorithm depends on the size of the input.
- When the input grows, also the running time grows.
- Let us denote by $n$ the size of the input. For instance, if an algorithm has a list as its input, the size $n$ is the number of the elements in the list.
- We denote by $T(n)$ the running time of the algorithm with size $n$ input.

# Running time of a loop

For instance, in the loop

```
for i in range(n):
    k = k + i
```

the size of the "input" is *n*. This is because the loop runs *n* times, and the variable *k* is updated in each iteration.

We can think that each addition `k = k + i` takes a constant time $c$. Therefore, the running time of the algorithm is

$$T(n) = cn$$

# Loop inside a loop

Similarly, the running time of

```
for i in range(n):
    for j in range(n):
        k = k + i + j
```

is $T(n) = c \cdot n^2$

# A practical test

```python
import time

n = 1000
k = 0

start_time = time.time()    # Current time in seconds since the epoch

for i in range(n):
    for j in range(n):
        k = k + i + j


end_time = time.time()

running_time = end_time - start_time
print(f"With the size of n = {n}, the running time is {running_time:.3f} seconds")
```

**File**: `two_loops.py`

# Running times

- With the size of n = 1 000, the running time is 0.070 seconds
- With the size of n = 10 000, the running time is 8.211 seconds
- With the size of n = 50 000, the running time is 192.962 seconds

- The running time starts to grow very fast. This means that an algorithm which contains loops inside a loop will be non-usable very fast.
- With three nested loops the situation is much worse.

# Grow rate of functions

| $n$ | $n \lg(n)$ | $n^2$ | $n^3$ |
|---|---|---|---|
| 10 | 33.22 | $10^2$ | $10^3$ |
| 100 | 664.39 | $10^4$ | $10^6$ |
| 1000 | 9965.78 | $10^6$ | $10^9$ |
| 10000 | 132877.12 | $10^8$ | $10^{12}$ |
| 100000 | 1660964.39 | $10^{10}$ | $10^{15}$ |

In the above table, $lgn$ denotes the denotes the base-2 logarithm of $n$.