- 1 - Algorithm analysis

This section will focus on the analysis of algorithms in terms of complexity. To be more precise, even though we should also focus on the space complexity, the following discussion will just refer to the time one. Also, we will make a focus on the *amortized time complexity*.

To start, what do we mean by time complexity? Basically we refer to how well (or bad) our algorithm does in terms of time, or put in other words, how much time it needs to complete its execution.

We can distinguish three cases, respectively:

1. *the best case:* the time required by the best input configuration;
2. *the average case:* the time required by the average input configuration;
3. *the worst case:* the time required by the worst possible input configuration.

Remark: From now on we will refer to each cases using the associated notation. Briefly, we will use Ω, Θ, $\mathcal{O}$ to refer to the best, average and worst case respectively.

As an example, let's consider the following code.

```
BinarySearch(array, target):
 low = 0
 high = length of array - 1

 while low <= high:
     mid = (low + high) / 2

     if array[mid] == target:
         return mid    // Target found, return the index
     else if array[mid] < target:
         low = mid + 1  // Target is on the right half, adjust low
     else:
         high = mid - 1  // Target is on the left half, adjust high

 return -1  // Target not found, return -1
```