17.1 Introduction to Average-Case Running Time

In our study of running-time analysis so far, we've divided up algorithms based on whether their running time depends on only the size of their inputs, or whether their running time depends on the actual value of the inputs as well, causing a spread of running times for any fixed input size. In the latter case, we've been concerned with the maximum boundary of this spread, which we call the *worst-case running time*.

However, in practice worst-case running time analysis can be misleading, with a variety of algorithms having a poor worst-case performance still yet performing well on the vast majority of inputs. Some reflection makes this not too surprising. Focusing on the maximum of a set of numbers says very little about the "typical" number in that set, or, more precisely, nothing about the *distribution* of numbers in that set.

One famous example we touched on in the previous section is the quicksort algorithm. Our implementation (and most implementations in practice) has a worst-case running time of $\Theta(n^2)$, but in practice this algorithm runs faster than mergesort on most inputs, even though the latter has a running time of $\Theta(n \log n)$. We said that the *average-case running time* of quicksort is $\Theta(n \log n)$, and in this chapter, we'll see precisely what this means.[1]

[1] Though we won't be looking at quicksort, but instead restricting ourselves to simpler examples. You'll study the average-case running time of quicksort in CSC263/CSC265!

In this section, we'll introduce a formal definition of average-case running time, and then in the next section we'll look at a few examples of analysing the average-case running time of a function.

Running time sets, worst-case, and average-case

First, let's recall some definitions from Chapter 8. Let `func` be a program. For eacn $n \in \mathbb{N}$, we can define the set $\mathcal{I}_{\texttt{func},n}$ to be the set of all inputs to `func` of size n. From this set, we can define the *worst-case running time* of `func` as the function

$$WC_{\texttt{func}}(n) = \max \left\{ \text{running time of executing } \texttt{func}(x) \mid x \in \mathcal{I}_{\texttt{func},n} \right\}$$

In English, we can say that $WC_{\texttt{func}}(n)$ is the maximum running time of `func` on an input of size n.

We define the **average-case running time** of `func` in a similar way, except now taking the average rather than maximum of the set of running times. Formally, the average-case running time of `func` is defined as the function $Avg_{\texttt{func}} : \mathbb{N} \to \mathbb{R}^+$, where

$$Avg_{\texttt{func}}(n) = \frac{1}{|\mathcal{I}_{\texttt{func},n}|} \times \sum_{x \in \mathcal{I}_{\texttt{func},n}} \text{running time of } \texttt{func}(x)$$

Notice that when there is only one input per size, or when all inputs of a given size have the same running time, the average-case running time is the same as the worst-case running time, as there is no spread. However, as we'll see over the next few sections, where there is a spread of running times, we need to take them all into account when calculating this average.