

1. While (or until) the condition is satisfied, repeat:
 - 1.1. Do this.
 - 1.2. Do that.
2. Carry on.

or when one or more steps are to be performed repeatedly as a variable v steps through a sequence of values:

1. For $v =$ sequence of values, repeat:
 - 1.1. Do this.
 - 1.2. Do that.
2. Carry on.

2.2 Efficiency of algorithms

Given an algorithm, we are naturally interested in discovering how efficient it is. Efficiency has two distinct facets:

- **Time efficiency** is concerned with how much (processor) time the algorithm requires.
- **Space efficiency** is concerned with how much space (memory) the algorithm requires for storing data.

Often we have a choice of different algorithms that solve the same problem. How should we decide which of these algorithms to adopt? Naturally we tend to prefer the most efficient algorithm.

Sometimes one algorithm is faster, while an alternative algorithm needs less space. This is a classic space–time tradeoff, which can only be resolved with knowledge of the context in which the chosen algorithm will be used.

In this book we shall tend to pay more attention to time efficiency than to space efficiency. This is simply because time efficiency tends to be the critical factor in choosing between alternative algorithms.

Usually, the time taken by an algorithm depends on its input data. Figure 2.1 shows a hypothetical profile of two alternative sorting algorithms, showing how the time they take depends on n , the number of values to be sorted. Algorithm A is slightly faster for small n , but algorithm B wins more and more easily as n increases.

How should we measure an algorithm's time efficiency? Perhaps the most obvious answer is to use real time, measured in seconds. Real time is certainly important in many practical situations. An interactive program that takes two minutes to respond to a user input will quickly fall into disuse. An aircraft control system that takes 30 seconds to respond to an abnormal altimeter reading will be eliminated by natural selection, along with the unfortunate crew and passengers.

Nevertheless, there are difficulties in using real time as a basis for comparing algorithms. An algorithm's real time requirement depends on the processor speed as well on the algorithm itself. Any algorithm can be made to run faster by using a faster processor.