MIT 6.006 Introduction to Algorithms

Sebastian Nyberg

May 16, 2021

Contents

0 Overview

- Course Website

1 Algorithmic thinking, asymptotic complexity, peak finding

Links:

- Lecture
- Recitation

This lecture is an introduction to algorithms in general.

1.1 Asymptotic complexity

Time complexities:

- Upper-bound: $\mathrm{O}(\mathrm{f}(n)) = \mathrm{g}(n)$, where $\mathrm{g}(n)$ is the **highest** possible value of f(n) as $n \to \infty$
- Lower-bound: $\Omega(\mathrm{f}(n)) = \mathrm{g}(n)$, where $\mathrm{g}(n)$ is the **lowest** possible value of f(n) as $n \to \infty$
- Average: $\Theta(\mathrm{f}(n)) = \mathrm{g}(n)$, where $\mathrm{g}(n)$ is the **most likely** value of f(n) as $n \to \infty$

Given

$$\mathrm{f}(n) = \log_{\ln 5}\left((\log n)^{100}\right)$$

What is the time complexity?

$$\log n^a = a \cdot \log n$$

Gives

$$\mathrm{f}(n) = 100 \cdot \log_{\ln 5}(\log n)$$

Since the base does not matter in asymptotic complexity ($\lim(\mathrm{n}) \to \infty$), and neither does the constant 100, the result is:

$$\Theta(\mathrm{f}(n)) = \log\log n$$

2 Models of Computation, Document Distance

This lecture talks about different ways of reasoning about computation. It also introduces the concept of document distance, which can be used to determine the likeness of documents.

- Lecture

2.1 Models of Computation

When reasoning about programs, there's a duality of described and actual. The design of algorithms does not concern itself with the actual, but rather with a description of a solution along with its most important characteristics. It is then up to the programmer to write a program, in a programming language, which runs on a computer.

To deal with how a computer actually works with the instructions that are part of an algorithm, something called a model of computation has been formed. These models allow algorithm designers to reason about how long it will take to perform an action, even if the actual hardware is unknown. This is very much similar to the lumped-circuit abstraction which ignores Maxwell's laws in favour of simple heuristics that work in the vast majority of cases.

There two most common models of computation are: Random Access Machine (RAM), and Pointer Machine. A RAM is visualized as one large chunk of memory consisting of cells. Each cell contains a word of data (typically 32 or 64 bits wide). A Pointer Machine deals with objects, and can point to other objects as needed. In a pointer machine, there is no need to consider the layout of memory.

2.2 Document Distance

A document is a sequence of words surrounded by whitespace and stopwords. A word is a sequence of alphanumerical symbols. The document distance is a measure of likeness between two documents.

One way to model words is to use a word vector. The most simple form of word vector is a frequency count of each word in the english language found in a document.

Comparing two documents is then a matter of comparing vectors.

One could imagine the use of dot-product (inner product). However, dot-products are not normalized, so the larger the text, the larger the dot-product.

A more efficient method is to use cosine-similarity - a measure of the angle between two vectors. This measure is given by

$$\cos\alpha = \frac{\mathrm{D}_1 \cdot \mathrm{D}_2}{|\mathrm{D}_1|\,|\mathrm{D}_1|}$$

This gives a normalized value from 0 to 1, depending on the similarity of the two vectors.