

Queen's University
School of Computing

CISC 203: Discrete Mathematics for Computing II
Assignment 1

Due January 31, 2019 at 11:30am

- [5 marks] 1. In algorithm analysis, we care a lot about how many steps ($f(n)$) an algorithm performs relative to the size of its input (n). For instance, an algorithm with a worst-case performance of n^3 is slower than one with a worst-case performance of n^2 , for large-enough n . In general, if an algorithm performs polynomially-many steps, then its performance worsens as the exponent grows. We can show a similar result for algorithms that perform exponentially-many steps; that is, x^n steps for some fixed x .
- Prove that if a and b are real numbers such that $0 < a < b$, then $a^n < b^n$ for all natural numbers $n \geq 1$.
- [5 marks] 2. UTF-8 is part of the Unicode character encoding standard. It is the most common standard for encoding text on the web, in email, and in documents.
- (a) UTF-8 is designed to be backwards-compatible with the ASCII encoding, which contains English alphabet characters and punctuation symbols. Characters in these alphabets have UTF-8 encodings of the form `0xxxxxxx`, where `x` is one bit (0 or 1). How many possible characters can be represented with this encoding?
 - (b) Other alphabets, like Greek and Russian, have UTF-8 encodings of the form `110xxxxx10xxxxxx`, where `x` is defined as before. How many possible characters can be represented with this encoding?
 - (c) The English alphabet has 26 uppercase letters, the Greek alphabet has 24 uppercase letters, and the Russian alphabet has 33 uppercase letters. Some of these alphabets share letters that look identical, so they don't need to be encoded multiple times. English and Greek share 14 letters, English and Russian share 12 letters, Greek and Russian share 14 letters, and all three languages share 11 letters. How many encodings in total do these three languages require, if shared letters also share encodings?
- [5 marks] 3. Imagine we are developing a new algorithm to optimize access to files stored in a computer's memory. Assume, for simplicity's sake, that our memory is divided into blocks of equal size, and each file we store takes up exactly one block of memory.
- (a) Version 1 of our algorithm simply takes a file and inserts it into an empty location in memory. If our computer has 10 empty memory locations, and we want to store 10 different files, in how many ways can our algorithm store every file?
 - (b) Version 2 of our algorithm stores all files of the same type together, so files of the same type are beside each other in memory. If our computer has 10 empty memory locations, and we want to store 4 text files, 3 music files, and 3 video files (all of which are different), in how many ways can our algorithm store every file?
- [5 marks] 4. Some text compression algorithms rely on the ability to condense blocks of consecutive identical bits into smaller encodings. Naturally, such algorithms do not perform well on bit strings with no consecutive identical bits. Let's investigate a simple example of these "bad" bit strings.
- How many bit strings of length 8 contain no consecutive 0 bits?
- Hint.* Split your solution into cases, with each case considering all bit strings of length 8 containing exactly k 0 bits. Do not simply list all of the bit strings explicitly.