

Problem Set 8 for CS 170

Note

When asked for an algorithm you must give (1) a brief informal description of the algorithm, (2) a precise description using pseudo-code, (3) an argument for termination and correctness of the algorithm, and (4) an analysis of the running time of the algorithm. Be clear about what the input to the algorithm is, how you measure the size of the input, and what constitutes a “step” in your running-time analysis.

Problem 1. [Optimizing the number of queries] (6 points)

Google wants to process a webpage and find out what language it is written in. It has bought the latest technological marvel from Oracle, a knowledge database. One can ask any question which has a yes-or-no answer to this database, and it gives the right answer. Google knows that of all the webpages on the internet, 40% are English, 17% are German, 15% are French, 11% are Chinese, 9% are Hindi, 5% are Russian, 2% are C, and the remaining 1% are “other.”

- (a) How many oracle queries does Google need on average to figure out the language of a webpage if it does a linear search in order of decreasing frequencies (first asking the oracle if the page is English, then German, then French, etc., stopping at the first yes answer)?
- (b) Can you help Google develop a better scheme? (*Hint*: you may want to use “or” queries of the form “is it English or French,” which have yes-or-no answers.) What is the expected number of oracle queries Google would make on each webpage according to your scheme?

Problem 2. [How long can Huffman codewords get?] (6 points)

- (a) Prove that in the Huffman coding scheme, if some character occurs with frequency more than $2/5$, then there is guaranteed to be a codeword of length 1. Prove also that if no character occurs with frequency at least $1/3$, then there is guaranteed to be no codeword of length 1.
- (b) Codex Unlimited manufactures hardware to decode Huffman-encoded files in an Xbox when you play over the network. Its decoding hardware needs to store an entire codeword (the bit sequence that encodes a character) in a register when it looks for a match. The uncompressed game files contain characters from an alphabet of size n . What is the minimum required register width (number of bits) that can support all Huffman codes for n characters, no matter what the frequencies f_1, \dots, f_n are? What relationship between the frequencies f_1, \dots, f_n causes the worst case (longest codeword) to occur?

Problem 3. [Ternary Huffman] (6 points)

TriMedia Disks Inc. has developed ternary hard disks. Each cell on a disk can now store values 0, 1, or 2. To take advantage of this new technology, provide a modified Huffman algorithm for compressing sequences of characters from an alphabet of size n , where the characters occur with known frequencies f_1, \dots, f_n . Your algorithm should achieve maximal possible compression by encoding each character with a variable-length codeword over the letters 0, 1, and 2 such that no codeword is a prefix of another codeword. Prove your algorithm correct.

Problem 4. [Lempel-Ziv] (6 points)

- (a) Alice sent Bob a Lempel-Ziv encoded file. She forgot to send Bob the dictionary. Bob knows that Alice uses k bits to represent her dictionary indices (codewords), and that she uses a dictionary of size n . Can Bob decode the file? If so, how?
- (b) Assume that we use Lempel-Ziv with a dictionary of unbounded size. If we encode a binary sequence of 28 bits, how large can the dictionary get in the worst case? How small can our dictionary be in the best case? Give one of the worst-case and one of the best-case input sequences.

Problem 5. [Linear Programming] (6 points)

- (a) Find necessary and sufficient conditions for the reals a and b to make the linear-programming problem “maximize $x_1 + x_2$ subject to $ax_1 + bx_2 \leq 1$ and $x_1 \geq 0$ and $x_2 \geq 0$ ” (1) be infeasible (have no solution), (2) be unbounded (for every solution there is a better one), and (3) have a unique optimal solution. Is there a fourth possibility?
- (b) Can you write a set of linear constraints on two real-valued variables x_1 and x_2 , and a quadratic objective function, such that the feasible region is nonempty and bounded, but none of the vertices of the feasible region is the optimal solution?