Homework 9

CMPSC 465

Kinner Parikh

November 17, 2022

Problem 1:

I worked with Sahil Kuwadia and Ethan Yeung
I did not consult without anyone my group member
I did not consult any non-class materials

Problem 2: *Suppose we have an optimal prefix code on a set $C = \{0, 1, ..., n-1\}$ of characters and we wish to transmit this code using as few bits as possible. Show how to represent any optimal prefix code on C using only $2n - 1 + n\lceil \log n \rceil$ bits.*

Since there are n characters, there are n leaves in the tree. Thus, there will be $n - 1$ vertices within the graph, so the entire graph will contain $2n-1$ total vertices, thus $2n-1$ bits. The height of a full binary tree for n characters is $\lceil \log n \rceil$.

We can say that to associate the members of C with the leaves of the tree, $\lceil \log n \rceil$ bits will be enough to represent all members. We know that no padding bits are required because each character is represented by a unique prefix. So, with n leaves, it requires $n\lceil \log n \rceil$ bits to represent all characters.

Thus, we can say that the total number of bits required to represent the optimal prefix code is $2n - 1 + n\lceil \log n \rceil$.

Problem 3: *Generalize Huffman's algorithm to ternary codewords (i.e., codewords using the symbols 0, 1, and 2), and prove that it yields optimal ternary codes.*

Algorithm 1: Huffman's Algorithm for Ternary Codes

Input: $f : f[1 \dots n]$
Output: T : a tree with n leaves

T: empty tree;
H: priority queue ordered by f;
for $i := 1$ ***in*** n **do**
  insert(H, i);
end for
for $k := n + 1$ ***in*** $\lceil 2n - n/2 \rceil$ **do**
  i := extract_min(H);
  j := extract_min(H);
  k := extract_min(H);
  Create a node r in T with children i, j, k;
  $f[r] = f[i] + f[j] + f[k]$;
  insert(H, r);
end for
return T

Claim: Huffman's algorithm for ternary codes yields an optimal ternary code. Every optimal solution has three least frequent symbols as leaves connected to an internal node of greatest depth

Proof: Let T be an optimal ternary code. Let i, j, k be the three least frequent symbols. Let r be the internal node of greatest depth. Let T' be the tree obtained from T by removing i, j, k and r. Let T'' be the tree obtained from T by removing r. Then T' is an optimal ternary code and T'' is an optimal binary code.

Problem 4: