الاسم : اسلام هاني عبد العزيز
شعبة : حاسب واحصاء

Text compression and Huffman coding

Text compression is useful to save many large documents in small storage space inside the computer, also reduce the time required to transfer any text when we are communicating over low-bandwidth channel, such as wireless or satellite connection, text is compressed by using standard encoding systems such as ASCII and Unicode systems (using two characters 0 and 1) and the most frequently used characters takes the fewest number of bits.

Huffman coding using to solve text compression problem, provide a method of encoding data efficiently. Normally, method produces a variable-length prefix code

The critical part of the Huffman coding algorithm is to construct the tree T, so that it represents a good prefix code. Each external node v in T is associated with a character and has a frequency,

Algorithm Huffman(C):

Input: A set, C, of d characters, each with a given weight, $f(c)$

Output: A coding tree, T, for C, with minimum total path weight

Initialize a priority queue Q.

for each character c in C do

Create a single-node binary tree T storing c.

Insert T into Q with key $f(c)$.

while Q.size() $>$ 1 do

$f1 \leftarrow Q$.minKey()

$T1 \leftarrow Q$.removeMin()

$f2 \leftarrow Q$.minKey()

$T2 \leftarrow Q$.removeMin()

Create a new binary tree T with left subtree $T1$ and right subtree $T2$.

Insert T into Q with key $f1 + f2$.

return tree Q.removeMin()

f(v), which is the frequency in X of the character associated with v. For each internal node, v, in T, we associate a total frequency, f(v), which is the sum of the frequencies of all the external nodes in the subtree rooted at v.

Each iteration of the while-loop in the Huffman coding algorithm can be implemented in O(log d) time using a priority queue represented with a heap. In addition, each iteration takes two binary trees out of Q and adds one in, all of which can be done in O(log d) time. This process is repeated d − 1 times before exactly one node is left in Q. Thus, this algorithm runs in O(d log d) time, assuming we are given the set, C, of d distinct characters in the string X as input.

Lemma: defined for each c in C, two characters, b and c, with the smallest two weights, are associated with nodes that have the maximum depth and are siblings in a binary tree T with minimum total path weight for C.

of the greedy method. this technique is applied to optimization problems, where we are trying to construct some structure while minimizing or maximizing some property of that structure Indeed, the Huffman coding algorithm closely follows the general formula for the greedy method. Namely, solve the given optimization

code problem, using the greedy method, we proceed by a sequence of choices. The sequence starts from a well-understood starting condition, and computes the cost for that initial condition.