

Assignment - 01
Advanced Data Structures and Algorithms
Indian Institute of Information Technology Kalyani

Due date: July 24, 2021

1. Implement the following procedures for Max-Heap management: BUILD-HEAP, HEAPIFY, REMOVE-MAX, INSERT-ELEMENT.
2. Implement a priority queue Q using Max-Heap. Show the following Insert and remove operations on Q in order as written below.
Insert: P, Q, A, D, X, C,
Remove Max
Insert: R, B, F, Y, G
Remove Max
Insert T, Z, U, L
Remove Max
3. Assume that the string “ab ab abcdefg” would be written as 103 111 32 103 111 32 103 111 112 104 101 114 115 numerically. This may be written as the following stream of bits:
1100111 1101111 0100000 1100111 1101111 0100000 1100111 1101111 1110000 1101000 1100101 1110010 1110011 (spaces do not exist).
Since there are only eight different characters in “ab ab abcdefg”, it’s possible to use only 3 bits to encode the different characters. For example, we may use the encoding in the table given below. Other encodings are also possible.

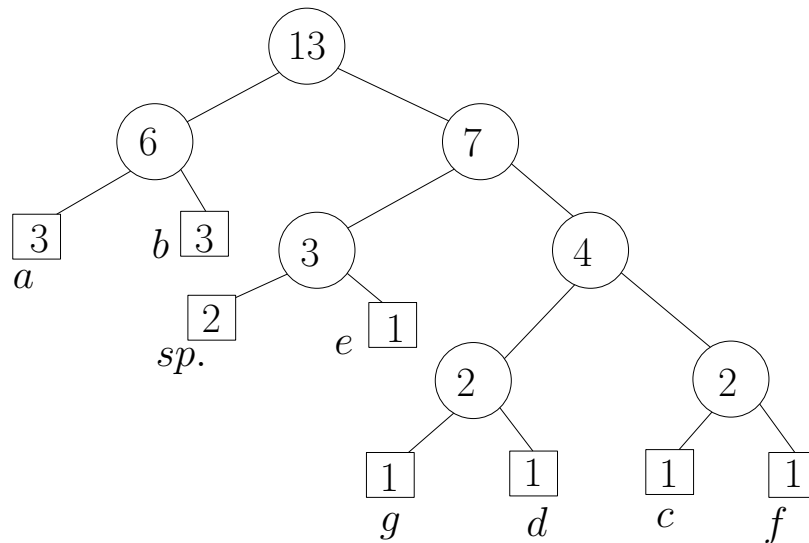
Character	Code
a	0
b	1
c	2
d	3
e	4
f	5
g	6
‘	7

Now the string “ab ab abcdefg” would be encoded as 0 1 7 0 1 7 0 1 2 3 4 5 6 or, as bits:

000 001 111 000 001 111 000 001 010 011 100 101 110 111 (spaces do not exist)

By using three bits per character, the string “ab ab abcdefg” uses a total of 39 bits instead of 104 bits. More bits can be saved if we use fewer than three bits to encode characters like *a*, *b*, and space that occur frequently and more than three bits to encode characters like *c*, *d*, *e*, *f*, and *g* that occur less frequently in “ab ab abcdefg”. This is the basic idea behind Huffman coding: to use fewer bits for more frequently occurring characters. We have discussed how this is done using a tree that stores characters at the leaves of the tree, and whose *root-to-leaf* paths provide the bit sequence used to encode the characters.

Prefix codes and Huffman Codes: When all characters are stored in leaves, and every interior/(non-leaf) node has two children, the coding induced by the 0 or 1 (left or right) ensures prefix property: no bit-sequence encoding of a character is the prefix of any other bit-sequence encoding. This makes it possible to decode a bitstream using the coding tree by following root-to-leaf paths. See the Huffman code tree and Huffman codes below for the above example.



The Huffman codes are:

Character	Code
a	00
b	01
c	1110
d	1101
e	101
f	1111
g	1100
'	100

Write a program for Huffman encoding of messages. During the construction of the Huffman tree, use priority queue for selection of partial trees to be merged.