#Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm.

import heapq

import time

from collections import defaultdict

start=time.time()

def encode(frequency):

heap = [[weight, [symbol, '']] for symbol, weight in frequency.items()]

print("\nHeap:",heap)

heapq.heapify(heap)

print("\nI am heap:",heapq)

while len(heap) > 1:

lo = heapq.heappop(heap)

print("\nLOW:",lo)

hi = heapq.heappop(heap)

print("HIGH:",hi)

for pair in lo[1:]:

pair[1] = '0' + pair[1]

for pair in hi[1:]:

pair[1] = '1' + pair[1]

heapq.heappush(heap, [lo[0] + hi[0]] + lo[1:] + hi[1:])

return sorted(heapq.heappop(heap)[1:], key=lambda p: (len(p[-1]), p))

data = ("MyString")

frequency = defaultdict(int)

for symbol in data:

frequency[symbol] += 1

print("\nFreq:",frequency)

huff = encode(frequency)

print ("\nSymbol".ljust(10) + "Weight".ljust(10) + "Huffman Code")

for p in huff:

print (p[0].ljust(10) + str(frequency[p[0]]).ljust(10) + p[1])

end=time.time()

diff=end-start

print("Time Complexity",diff)