(B) Huffman Encoding.

1. ALGORITHM

```
→ def compress (text):
    → freq= Counter(text)
    → If freq ==0:
          print ("Empty File")

    → htree = huffman_tree (freq)
    → val = codes (htree)

⇒ def huffman_tree (freq):
    → heapq.heapify (freq)

    → while heap.size >1:
          l = heapq.heappop()
          r = heapq.heappop()
          p = Node (l.freq + r.freq)
          p.left = l
          p.right = r

    → return root
coded = {}
⇒ def codes (root):
    → If root == None:
          return
      else
          coded[root.char] = val

    → codes (root.left, val+'0', coded)
    → codes (root.right, val+'1') coded)
→ def calc ()
    og_size = len (original_text) *8
    comp_size = len (compressed_text).

     ratio = og-size / comp-size,
     return ratio
```

Function to read Contents

```
→  def read (path):
        ~~doc = open~~
      with open (path, 'r') as file:
          text = file.read()
```

② Testcases.

· Positive Testcase.

- PDF File.
  Original content : This is a pdf file
  Compressed content: 0101100101.......
  Compression ratio: 1.9

- WORD Doc
  Original content : This is a word file
  Compressed Content : 010000110111110....
  Compression ratio: 1.7

- HTML File
  Original content : This is a HTML file
  ~~Comp~~ Compressed file : 0001101.....
  Compression ratio : 1.8

- TXT File
  Original Content : This is a text file.
  Compressed Content : 0011 0101......
  Compression ratio : 1.9

- MST Question
  Original : A wonderful serenity has taken possession
  of my soul like these sweet mornings of
  spring which I enjoy with whole heart.
  I am alone and feel the charm of existences in spot
  Compressed : 0111001110 0.....
  Compression ratio : 1.53.

Negative Testcases.

- Incorrect file path
  Display → No such file

- File is empty
  Display ⇒ The file is empty

- Not a PDF, HTML, DOCX, TXT File
  Display ⇒ Unsupported File Type.

- Library not installed
  Display ⇒ Module not present

3. Time Complexity.

Huffman encoding uses a heap data structure.

Initialising a heap takes O(n) time
Reading a file takes O(n) time
Operations on heap take O(nlogn) time
Calculating compression ratio takes O(1) time

$$\therefore T(n) = n + n + n\log n$$

$$= 2n + n\log n$$

$$T(n) = O(n\log n)$$