Lecture 6 (Huffman Encoding)

1 Formulation

Given: Symbols $(a_1, a_2, \cdots, a_n)$ with frequency vectors $F = (f_1, f_2, \cdots, f_n)$

Find: Prefix encoding such that the length of message is minimum, i.e., find a binary tree T with leaves $(a_1, a_2, \cdots, a_n)$, such that, $\sum_{i=1}^{n} f_i \cdot d(a_i)$ is minimum, where $d(leaf)$ is the depth of the leaf

Properties:

1. The tree will be complete
2. If $f_i \geq f_{i+1}$, then
    i. $d(a_i) \leq d(a_{i+1})$
    ii. $d(a_n) = d(a_{n-1})$

2 Algorithm

```
while len(alphabet) != 1:
    a1, a2 = letters with least frequency
    a_parent = Node(a1, a2) # a1, a2 are children of a_parent
    freq(a_parent) = freq(a1) + freq(a2)
    alphabet.remove(a1)
    alphabet.remove(a2)
    alphabet.add(a_parent)
```