CSE 3010 – Data Structures & Algorithms

Lecture #47

What will be covered today

- Huffman coding – Application of heap data structure

Huffman coding

- Developed by David Huffman in 1952
- Published a paper titled *'A Method for construction of minimum redundancy codes*'
- Algorithm that works with integer length codes
- Used to encode character in a file
- Huffman code is a special type of optimal prefix code
- Huffman code is used for lossless data compression
- Process of finding the optimal prefix code is called Huffman coding
- Output from Huffman algorithm is a variable-length code for encoding an input symbol

Understanding Huffman algorithm

Letter of the Alphabet	Fixed-length Code	Variable-length Code	Frequency Distribution
e	010	0	20
h	000	01	6
l	011	11	9
o	110	1	16

Word	Encoded Using Fixed-length Code	Encoded Using Variable-length Code
he	**000010**	**010 or 010**
hell	**000010011011**	**0101111 or 0101111**
hello	**000010011011110**	**Similarly we can have multiple combinations for hello**

How does Huffman algorithm work

- Input to the algorithm
  - Character set in the file - `{e, h, l, o}`
  - Frequency of each character in the file – `[20, 6, 9, 16]`
- Output of the algorithm
  - Variable-length code of the encoded input symbol(s)
- Steps [illustrated with the example in the class nores]:
  1. Pick the two symbols with the lowest frequencies `(h, l)`
  2. Make one of them the left child `(h-6)` and the other the right child `(l-9)`
  3. Add a root to the left and the right child – `(hl-15)`
  4. Remove the two symbols `(h, l)`, used to form the new root, from the character set
  5. Replace the character set with the root `(hl-15)` [this will have `h` and `l` as left and right children]
  6. Repeat Steps 1 to 5 until only one symbol is left in the character set