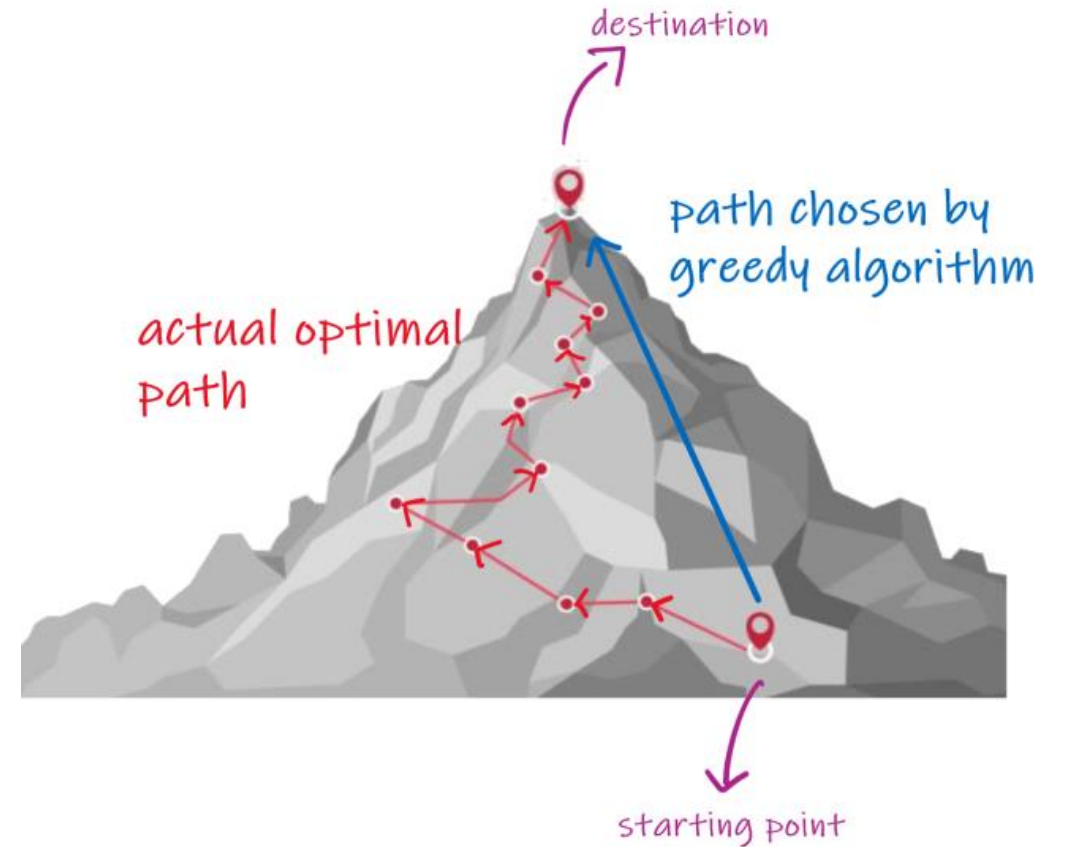


# DSE 2256 DESIGN & ANALYSIS OF ALGORITHMS

## Lecture 39 **Greedy Technique** Huffman Coding



# Coding Problem

**Coding:** Assignment of bit strings to alphabet characters

- E.g. We can code {a,b,c,d} as {00,01,10,11}

**Codewords:** Bit strings assigned for characters of alphabet

**Two types of coding:**

- Fixed-length encoding (e.g., ASCII)
- Variable-length encoding (e.g., Morse code, Huffman Code)

**Prefix-free codes (or prefix-codes):** no codeword is a prefix of another codeword

- It allows for efficient online decoding.

# Huffman codes

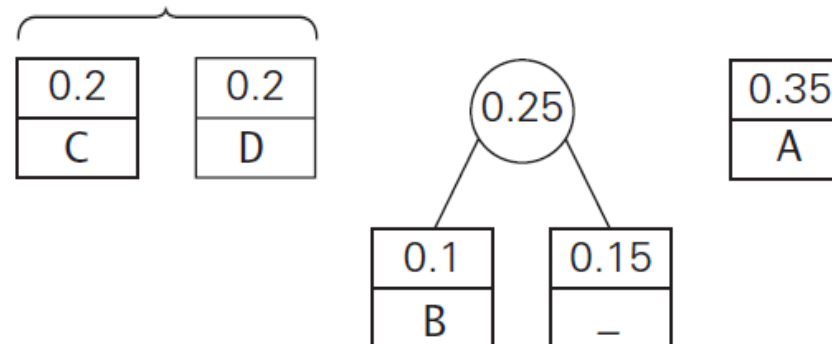
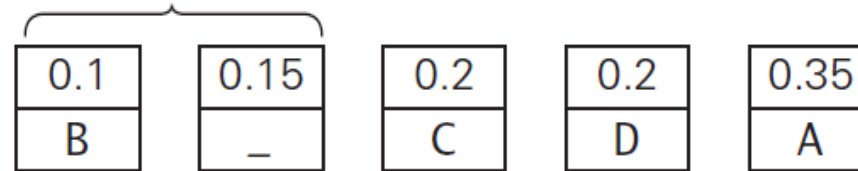
- Any binary tree with edges labeled with 0's and 1's yields a prefix-free code of characters assigned to its leaves.

## **Huffman's algorithm:**

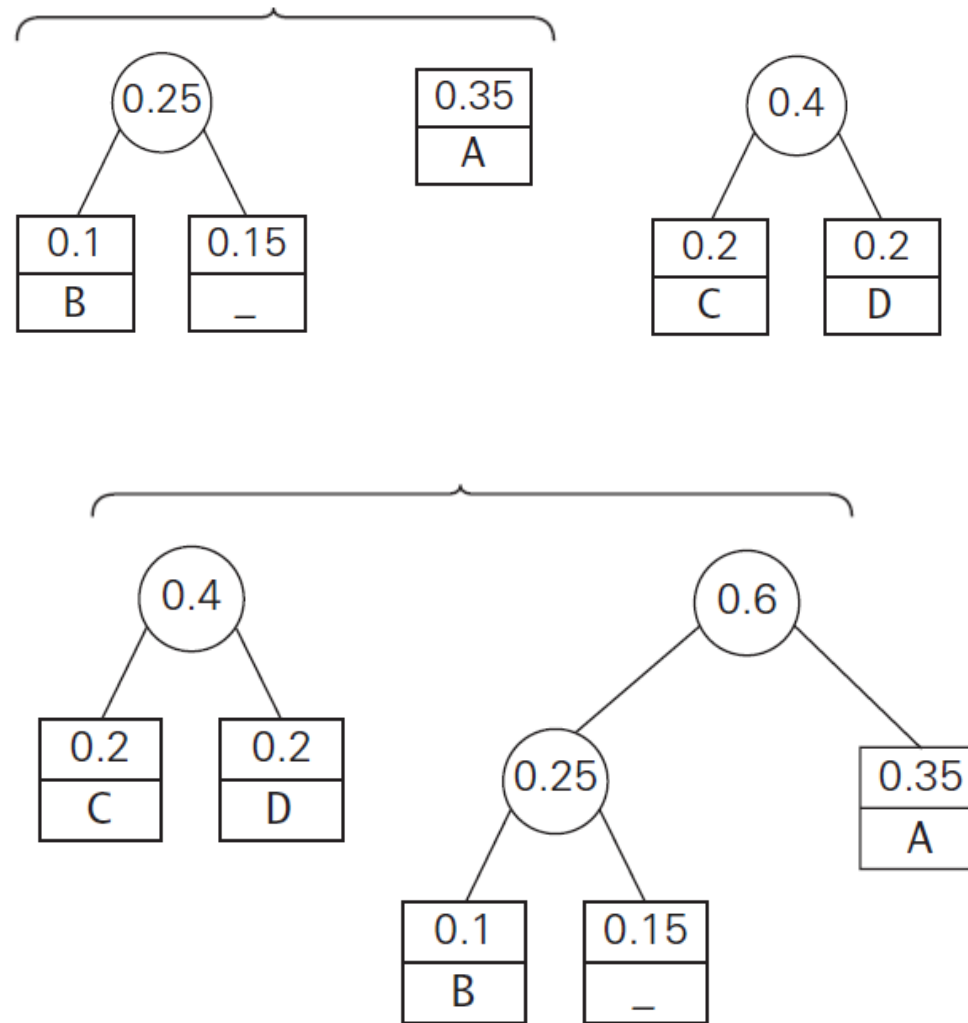
- Initialize  $n$  one-node trees with alphabet characters and the tree weights with their frequencies.
- Repeat the following step  $n-1$  times: join two binary trees with smallest weights into one (as left and right subtrees) and make its weight equal the sum of the weights of the two trees.
- Mark edges leading to left and right subtrees with 0's and 1's, respectively.

# Huffman codes : Example

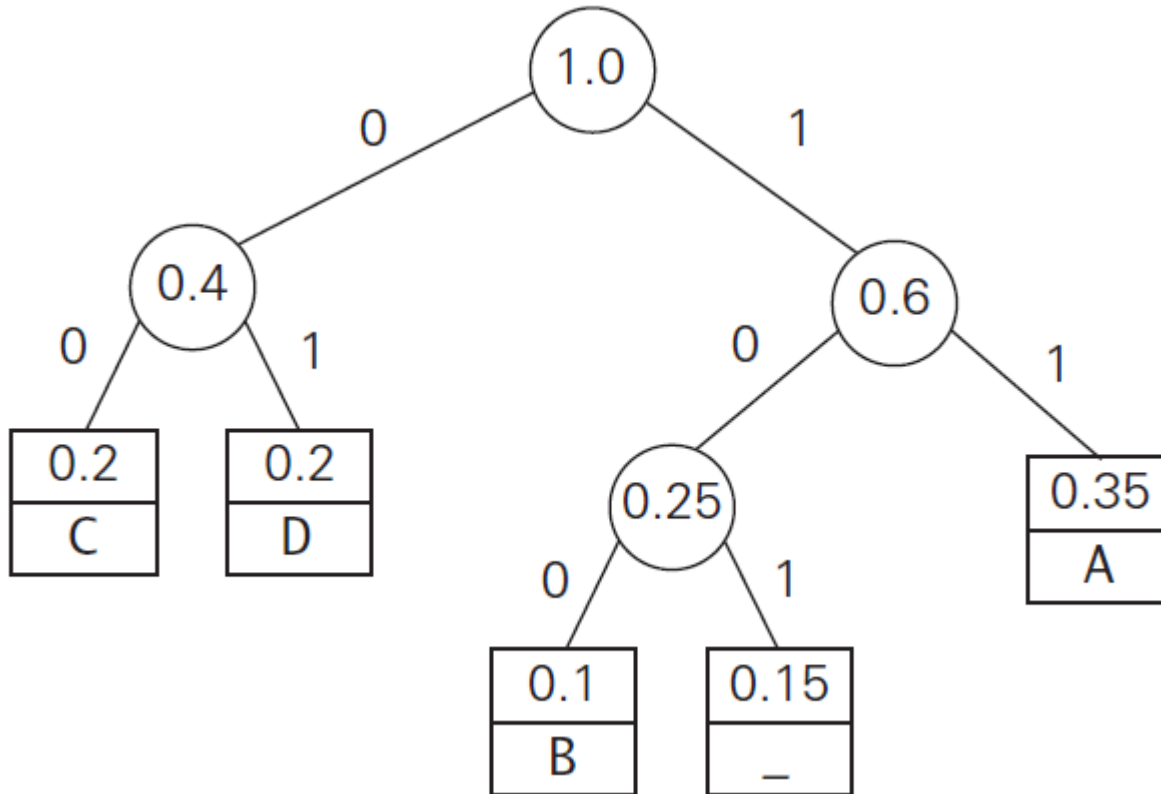
symbol	A	B	C	D	_
frequency	0.35	0.1	0.2	0.2	0.15



# Huffman codes : Example



# Huffman codes : Example



Symbol:	A	B	C	D	_
Frequency:	0.35	0.1	0.2	0.2	0.15
Codeword:	11	100	00	01	101

- Average bits per character: 2.25
- For fixed-length encoding: 3
- *Compression ratio:*  $(3 - 2.25) / 3 * 100\% = 25\%$

# Thank you!

## Any queries?