HUFFMAN CODING

PRESENTED BY:

TULASI DAHAL

M.E. COMPUTER

KATHMANDU UNIVERSITY, DHULIKHEL

Contents

Introduction

- Huffman Coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters.
- The most frequent character gets the smallest code and the least frequent character gets the largest code.
- Developed by David A. Huffman while he was a Ph.D. student at MIT and published in the 1952 paper "A Method for the Construction of Minimum Redundancy Codes".

Major Steps

- There are mainly three major steps in Huffman Coding:
  i. Prepare the frequency table
  ii. Build a Huffman Tree from input characters
  iii. Traverse the Huffman Tree and assign codes to characters

Huffman Tree

- Steps to build Huffman Tree :

i. Create a leaf node for each unique character

ii. Extract two nodes with the minimum frequency

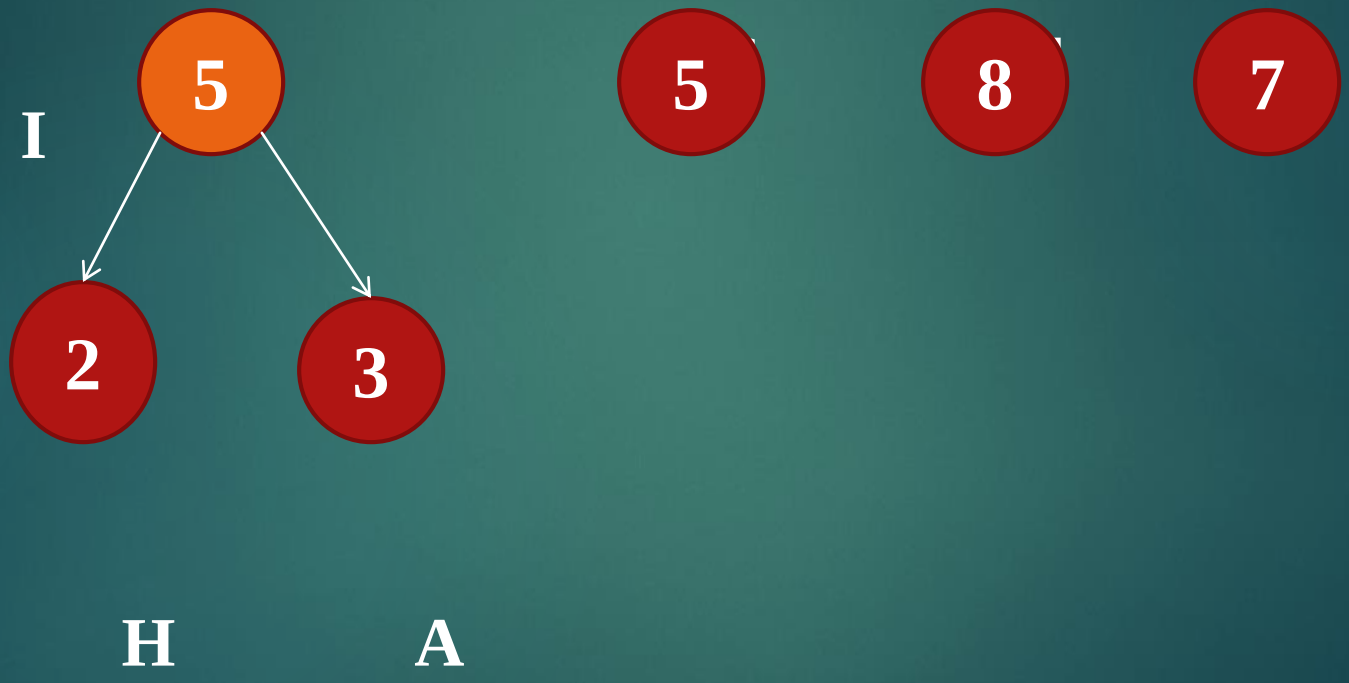
iii. Create a new internal node with frequency equal to the sum of the two nodes frequencies

iv. Repeat Step (ii) and (iii) until the heap contains only one node. The remaining node is the root node and the tree is complete.

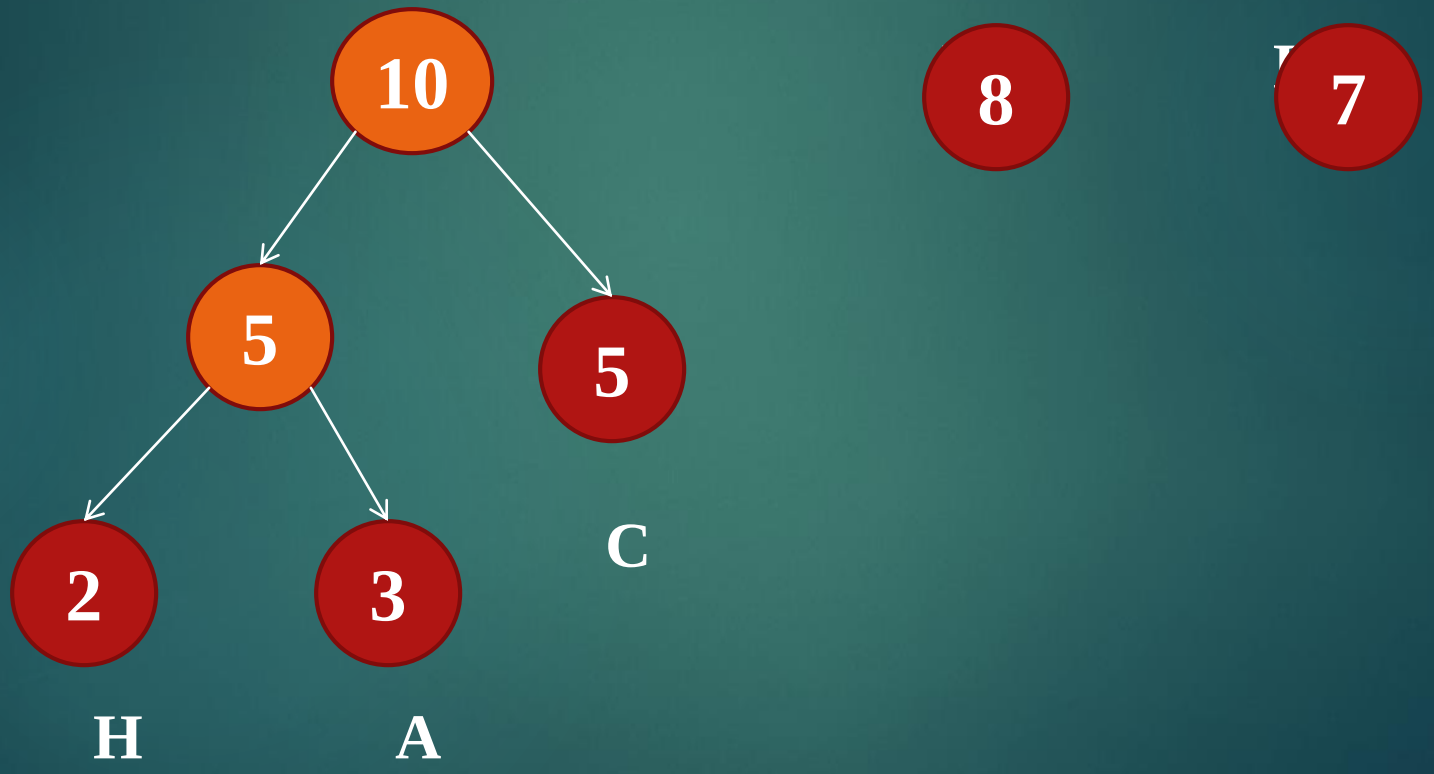
Huffman Tree (contd..)

- Example:

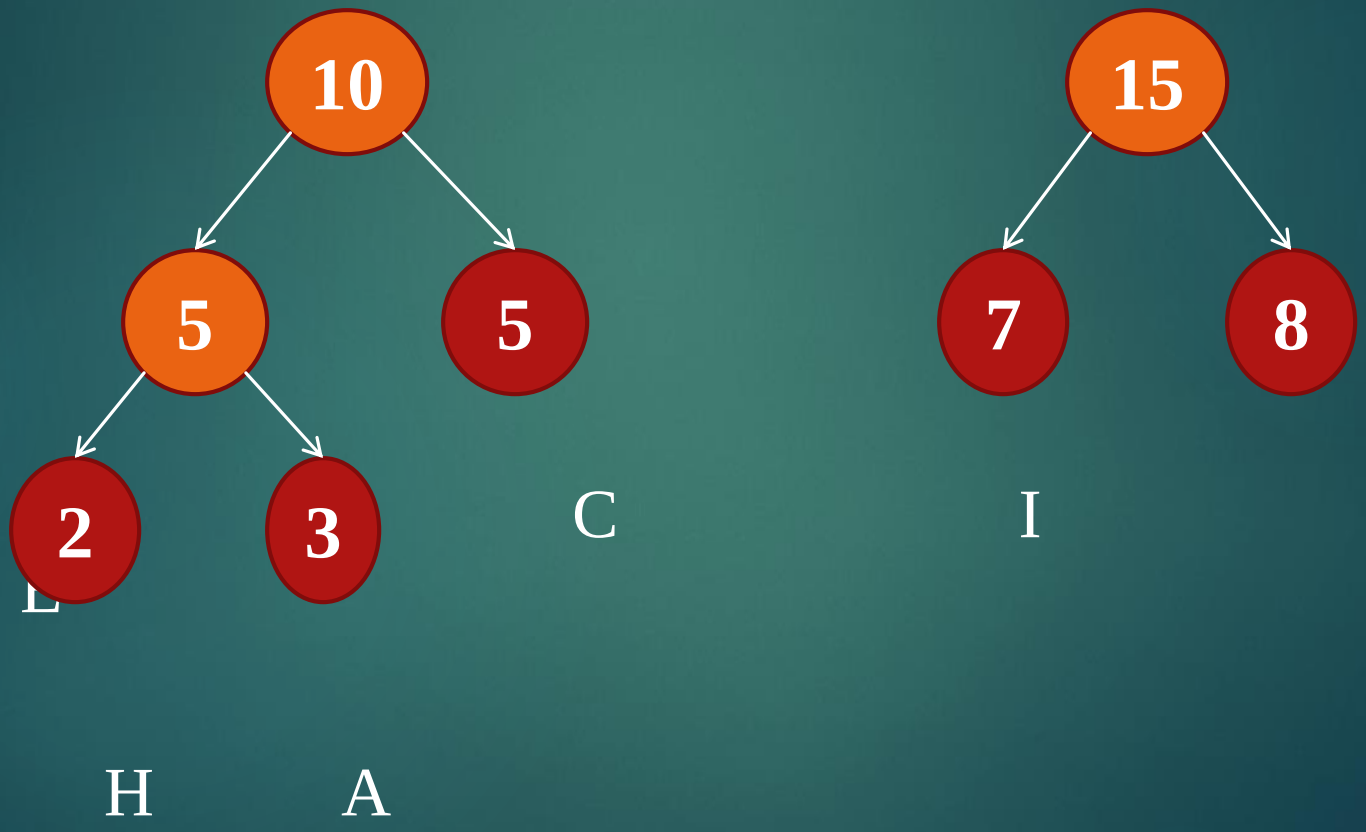
Huffman Tree (contd..)



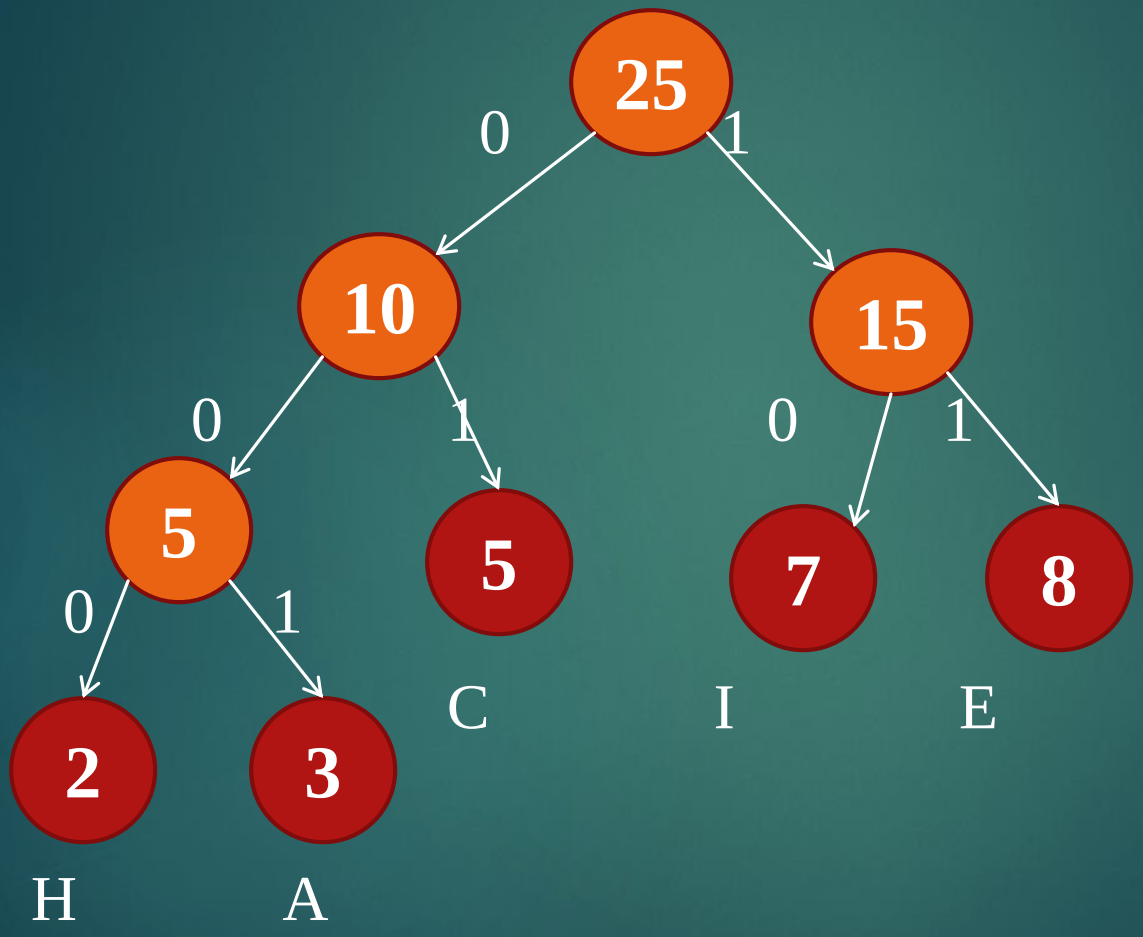
Huffman Tree (contd..)



Huffman Tree (contd..)



Huffman Tree (contd..)



A = 001

C = 01

E = 11

H = 000

I = 10

Algorithm

HUFFMAN(C)

1. n ← |C|
2. Q ← C
3. *for i* ← 1 *to n-1*
4. *do* allocate a new node z
5. x ← left[z] ← EXTRACT – MIN(Q)
6. y ← right[z] ← EXTRACT – MIN(Q)
7. *f* [z] ← *f* [x] + *f* [y]
8. INSERT (Q, Z)
9. *return* EXTRACT - MIN(Q)

Analysis

- Time Complexity:

  Time Complexity of Huffman Coding is O(n log n) where each iteration requires O(log n) time to determine the cheapest weight and there would be O(n) iterations.

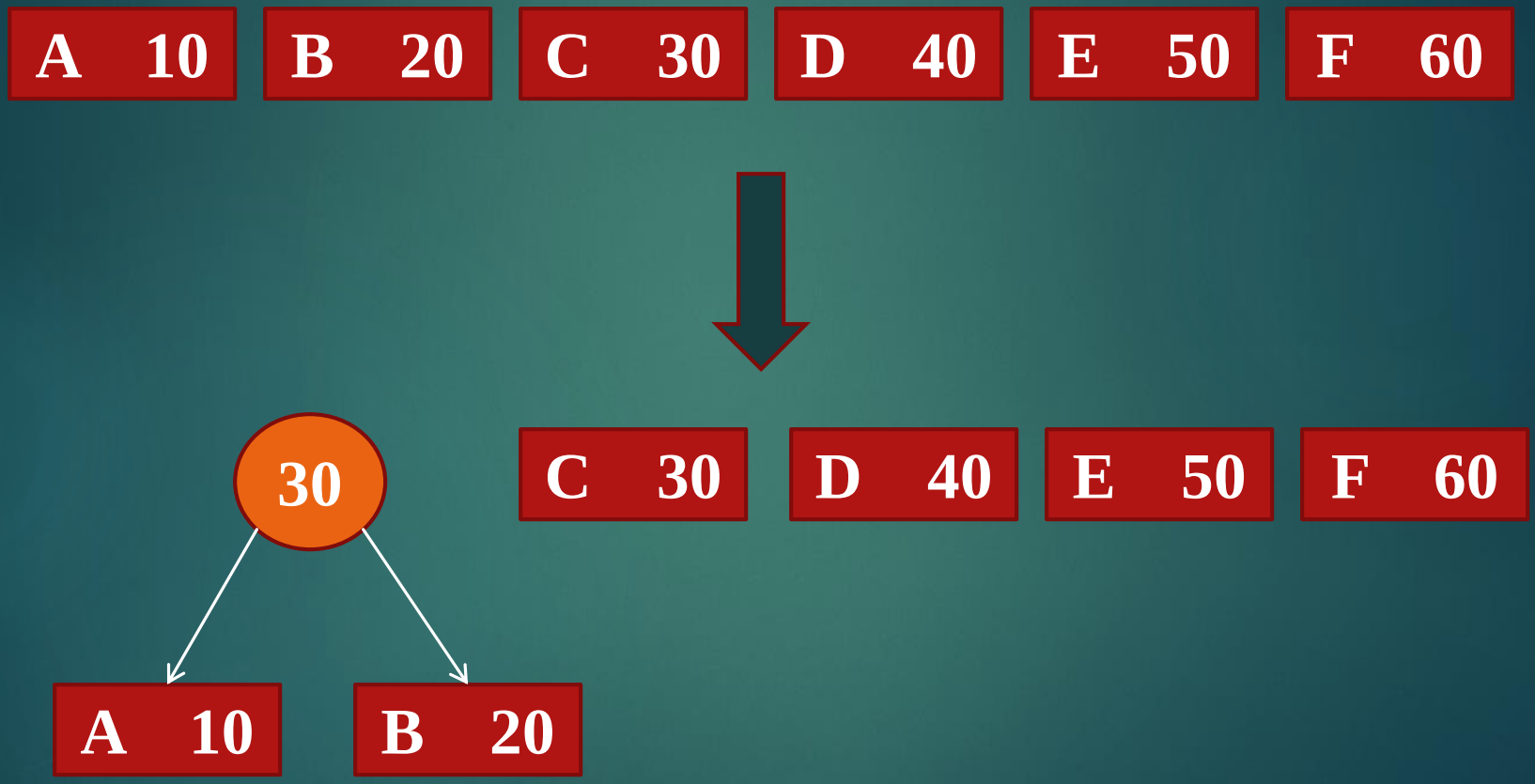
Worked Example

- Alphabet: A, B, C, D, E, F
- Frequency Table:

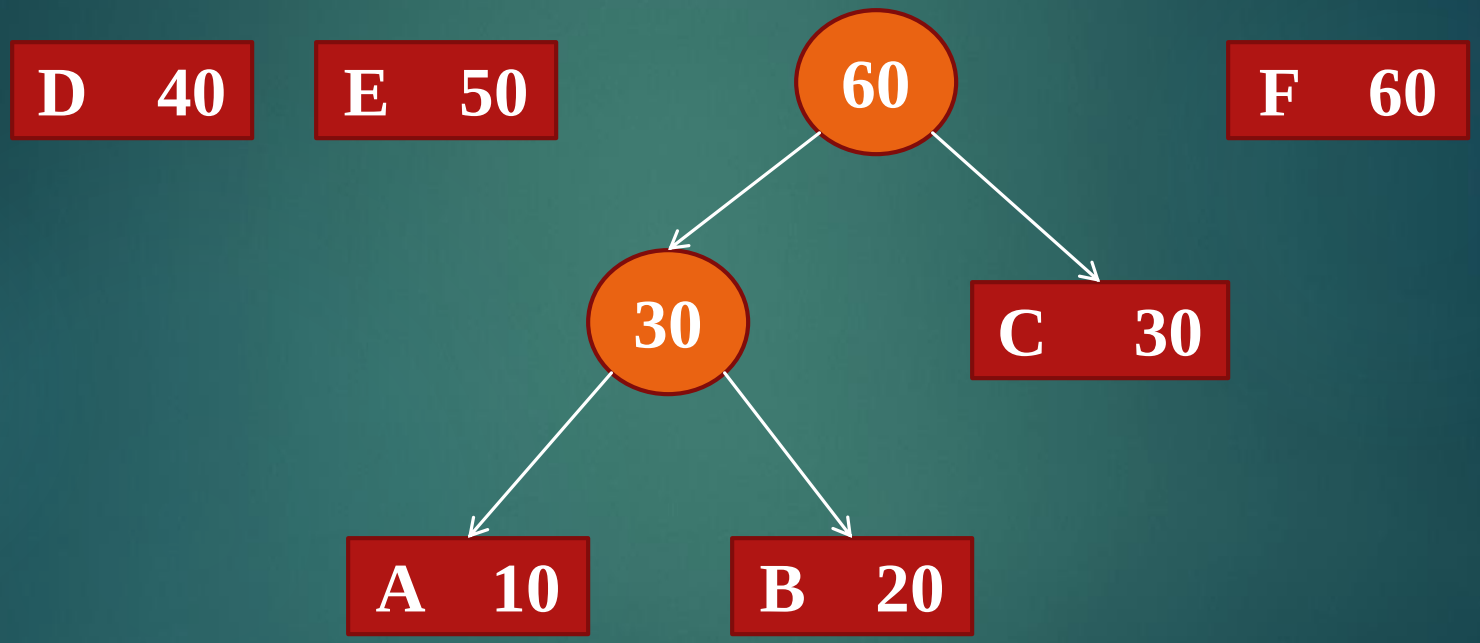
A	B	C	D	E	F
10	20	30	40	50	60

- Total File Length: 210

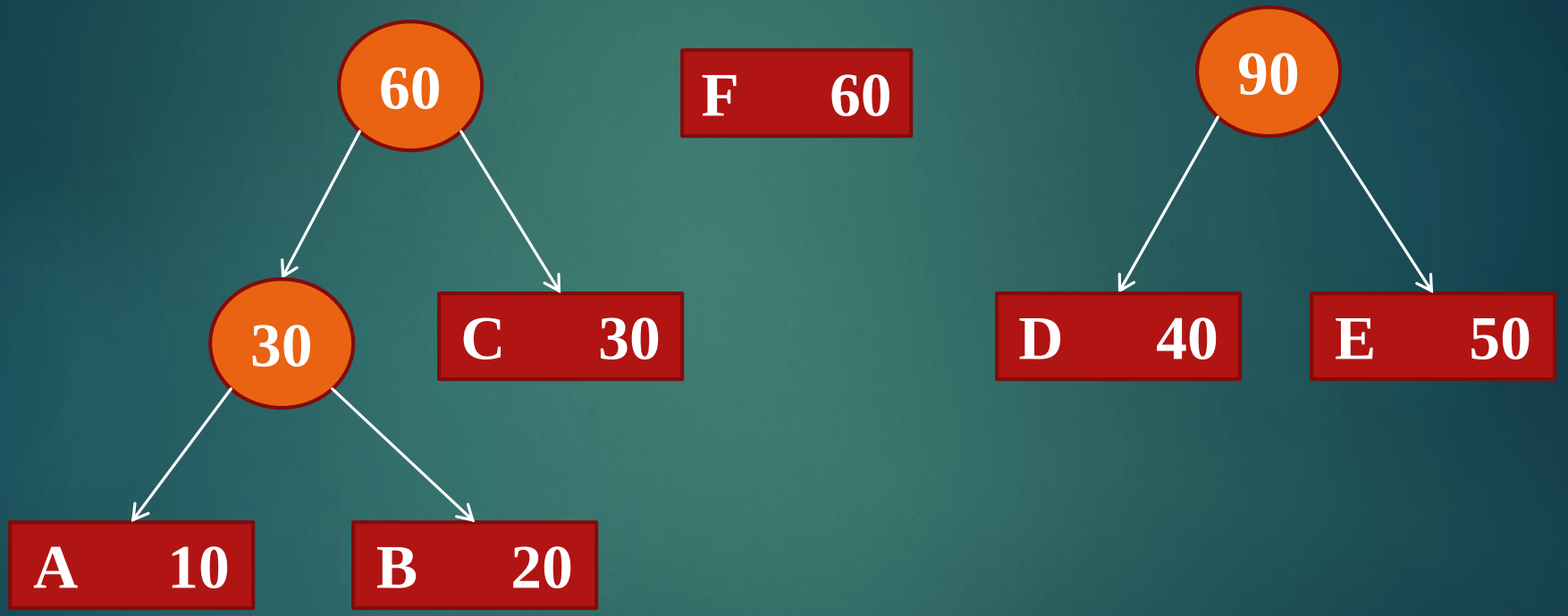
Worked Example (contd..)



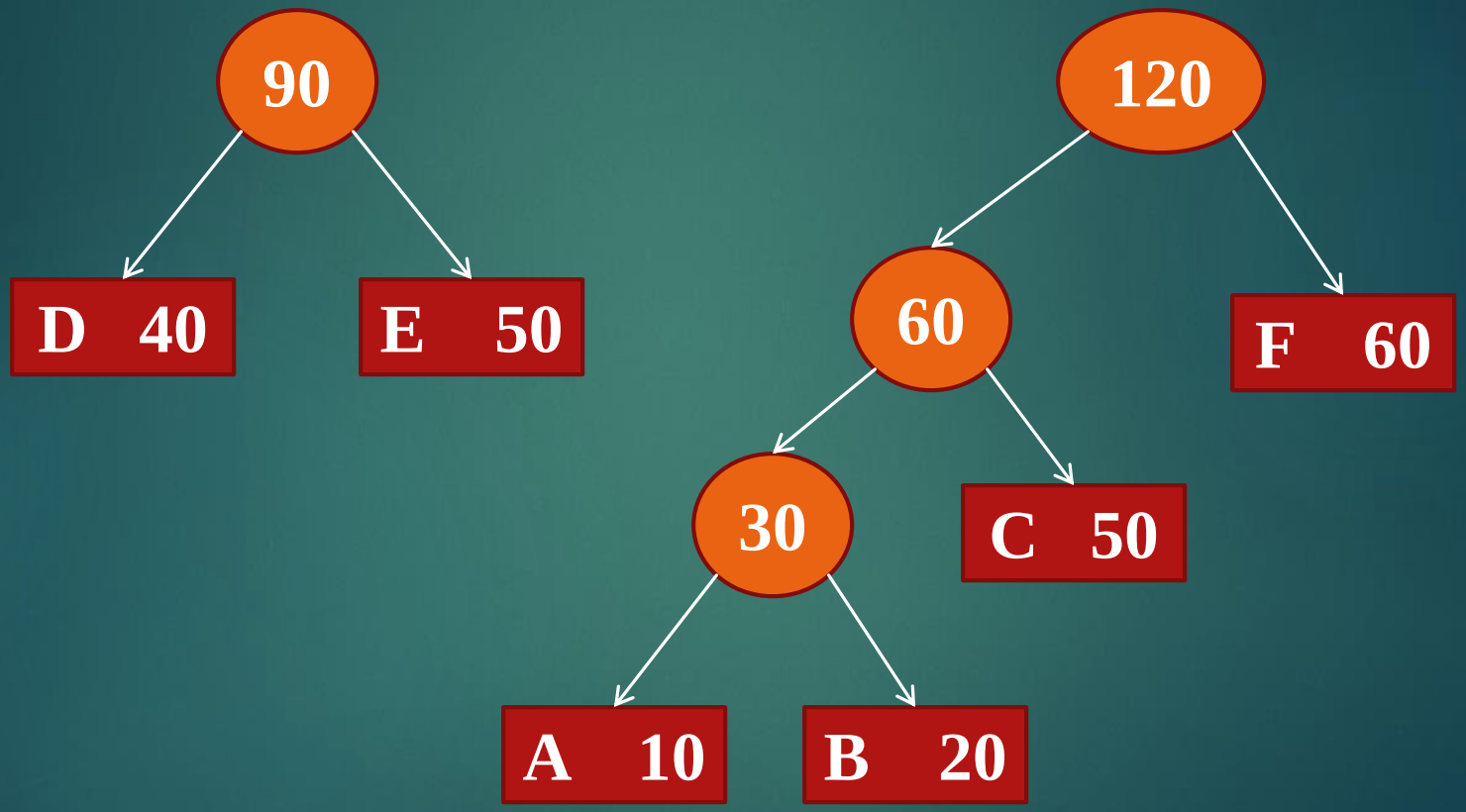
Worked Example (contd..)



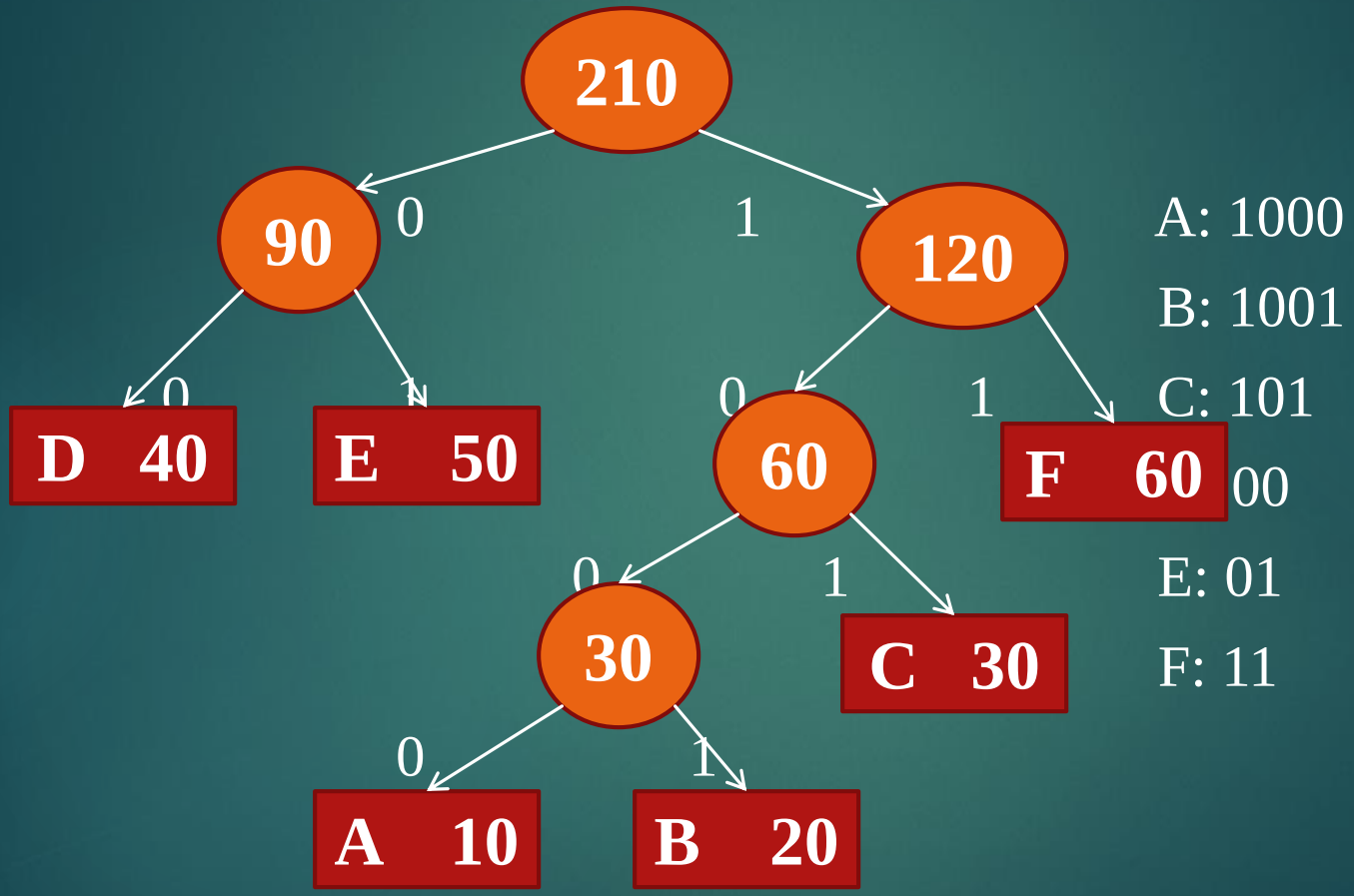
Worked Example (contd..)



Worked Example (contd..)



Worked Example (contd..)



A: 1000

B: 1001

C: 101

00

E: 01

F: 11

Worked Example (contd..)

Calculation:

- File Size = 10x4 + 20x4 + 30x3 + 40x2 + 50x2 + 60x2
  = 40 + 80 + 90 + 80 + 100 + 120 = 510 bits
- The Huffman Code:
  Required 510 bits for the file
- Fixed length code:

Need 3 bits for 6 characters

File has 210 characters

Total = 210x3 = 630 bits for the file

Properties of the Huffman Codes

- No Huffman code is prefix of any other Huffman codes so decoding is unambiguous
  - For example: In a given set of Huffman codewords, 10 and 101 can not simultaneously be valid Huffman codewords because first is the prefix of the second.
- The Huffman Coding technique is optimal i.e. savings of 20% to 90% are typical, depending on the characteristics of the file being compressed.
- Symbols that occur more frequently have shorter Huffman Codes

Conclusion

- Huffman Coding is a method for construction of minimum redundancy codes
- Also known as Probabilistic variable length coding
- It is widely used and very effective technique for compressing data
- Used in many compression algorithm like gzip, bzip, fax compression

THANK YOU!