

# Huffman coding

Problem:

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$
- ▶ **Optimal prefix code:** prefix code that minimizes the **weighted average number of bits per symbol**.

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$
- ▶ **Optimal prefix code:** prefix code that minimizes the **weighted average number of bits per symbol**.
  - ▶ **Example:**  $A:45$ ,  $B:16$ ,  $C:13$ ,  $D:12$ ,  $E:9$ ,  $F:5$

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$
- ▶ **Optimal prefix code:** prefix code that minimizes the **weighted average number of bits per symbol**.
  - ▶ **Example:**  $A:45, B:16, C:13, D:12, E:9, F:5$

**Encodings:**  $A:0, B:10, C:1100, D:1101, E:1110, F:1111$



# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$
- ▶ **Optimal prefix code:** prefix code that minimizes the **weighted average number of bits per symbol**.
  - ▶ **Example:**  $A:45, B:16, C:13, D:12, E:9, F:5$

**Encodings:**  $A:0, B:10, C:1100, D:1101, E:1110, F:1111$

**Weighted average symbol length:**

$$(45 \cdot 1 + 16 \cdot 2 + 13 \cdot 4 + 12 \cdot 4 + 9 \cdot 4 + 5 \cdot 4) / 100 = 2.33$$

# Huffman coding

## Problem:

- ▶ **Input:** Alphabet of symbols, frequency of each symbol.
- ▶ Want to encode each symbol as a sequence of bits
- ▶ **Prefix codes:** no symbol code is a prefix of another symbol code.
  - ▶ Can't have code for  $A = 00$ , code for  $B = 001$
- ▶ **Optimal prefix code:** prefix code that minimizes the **weighted average number of bits per symbol**.
  - ▶ **Example:**  $A:45, B:16, C:13, D:12, E:9, F:5$

**Encodings:**  $A:0, B:10, C:1100, D:1101, E:1110, F:1111$

**Weighted average symbol length:**

$$(45 \cdot 1 + 16 \cdot 2 + 13 \cdot 4 + 12 \cdot 4 + 9 \cdot 4 + 5 \cdot 4) / 100 = 2.33$$

(This encoding is not optimal)

# Huffman coding

Algorithm:

# Huffman coding

## Algorithm:

1. Build binary tree:

# Huffman coding

## Algorithm:

1. Build binary tree:
  - ▶ Start with alphabet symbols as nodes, no edges

# Huffman coding

## Algorithm:

1. Build binary tree:
  - ▶ Start with alphabet symbols as nodes, no edges
  - ▶ Find the two nodes without parents that have the smallest frequency.

# Huffman coding

## Algorithm:

1. Build binary tree:
  - ▶ Start with alphabet symbols as nodes, no edges
  - ▶ Find the two nodes without parents that have the smallest frequency.
  - ▶ Give them a common parent.

# Huffman coding

## Algorithm:

1. Build binary tree:
  - ▶ Start with alphabet symbols as nodes, no edges
  - ▶ Find the two nodes without parents that have the smallest frequency.
  - ▶ Give them a common parent.
    - ▶ It doesn't matter which is left child, which is right child



# Huffman coding

## Algorithm:

### 1. Build binary tree:

- ▶ Start with alphabet symbols as nodes, no edges
- ▶ Find the two nodes without parents that have the smallest frequency.
- ▶ Give them a common parent.
  - ▶ It doesn't matter which is left child, which is right child
  - ▶ Frequency of the parent is the sum of the frequency of the two nodes.

# Huffman coding

## Algorithm:

### 1. Build binary tree:

- ▶ Start with alphabet symbols as nodes, no edges
- ▶ Find the two nodes without parents that have the smallest frequency.
- ▶ Give them a common parent.
  - ▶ It doesn't matter which is left child, which is right child
  - ▶ Frequency of the parent is the sum of the frequency of the two nodes.
- ▶ Repeat until only one node has no parent (a tree)

# Huffman coding

## Algorithm:

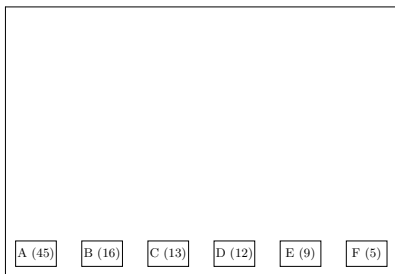
1. Build binary tree:
  - ▶ Start with alphabet symbols as nodes, no edges
  - ▶ Find the two nodes without parents that have the smallest frequency.
  - ▶ Give them a common parent.
    - ▶ It doesn't matter which is left child, which is right child
    - ▶ Frequency of the parent is the sum of the frequency of the two nodes.
  - ▶ Repeat until only one node has no parent (a tree)
2. Encode each symbol using through tree from the root to the symbol. (Left branch = 0, right branch = 1)

# Huffman Coding Example

Example:  $A:45$ ,  $B:16$ ,  $C:13$ ,  $D:12$ ,  $E:9$ ,  $F:5$

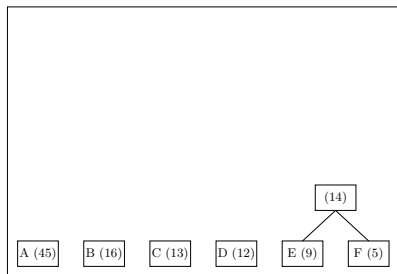
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



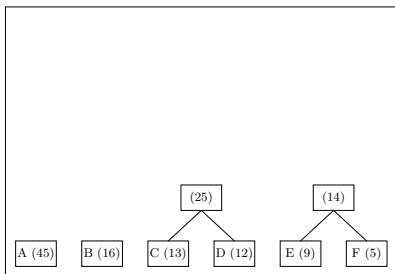
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



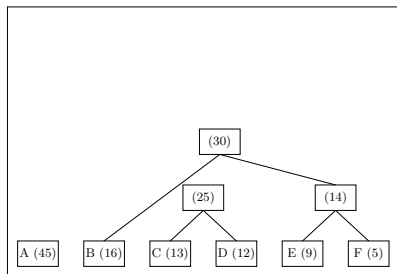
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



# Huffman Coding Example

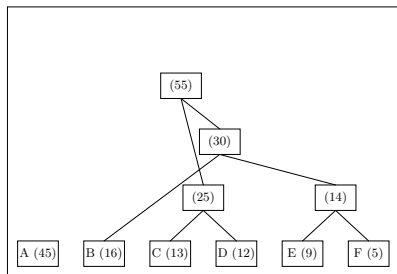
Example: A:45, B:16, C:13, D:12, E:9, F:5





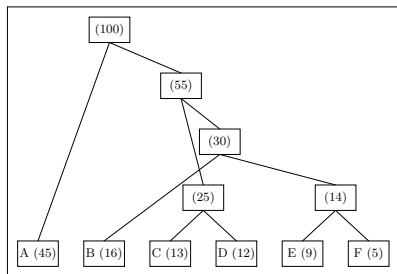
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



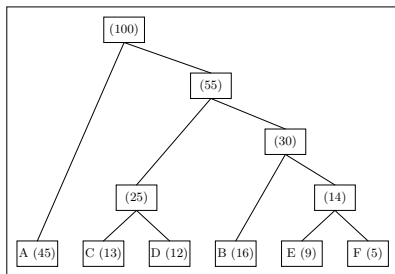
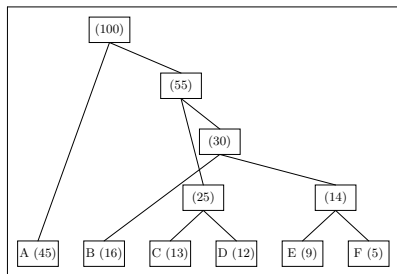
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



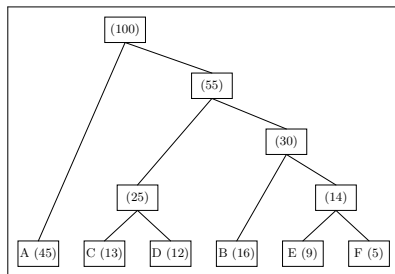
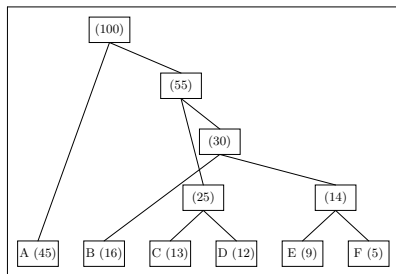
# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



# Huffman Coding Example

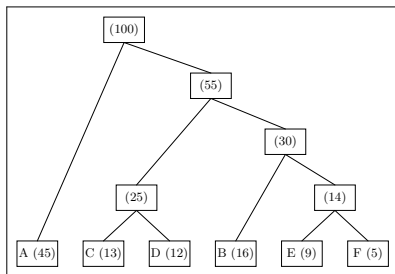
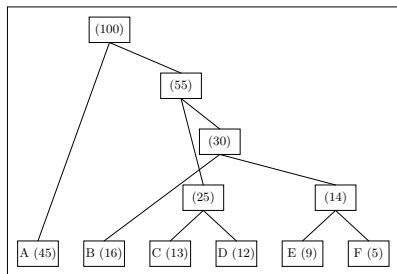
**Example:** A:45, B:16, C:13, D:12, E:9, F:5



**Encodings:** A:0, B:110, C:100, D:101, E:1110, F:1111

# Huffman Coding Example

Example: A:45, B:16, C:13, D:12, E:9, F:5



Encodings: A:0, B:110, C:100, D:101, E:1110, F:1111

Average symbol length:

$$(45 \cdot 1 + 16 \cdot 3 + 13 \cdot 3 + 12 \cdot 3 + 9 \cdot 4 + 5 \cdot 4)/100 = 2.24$$