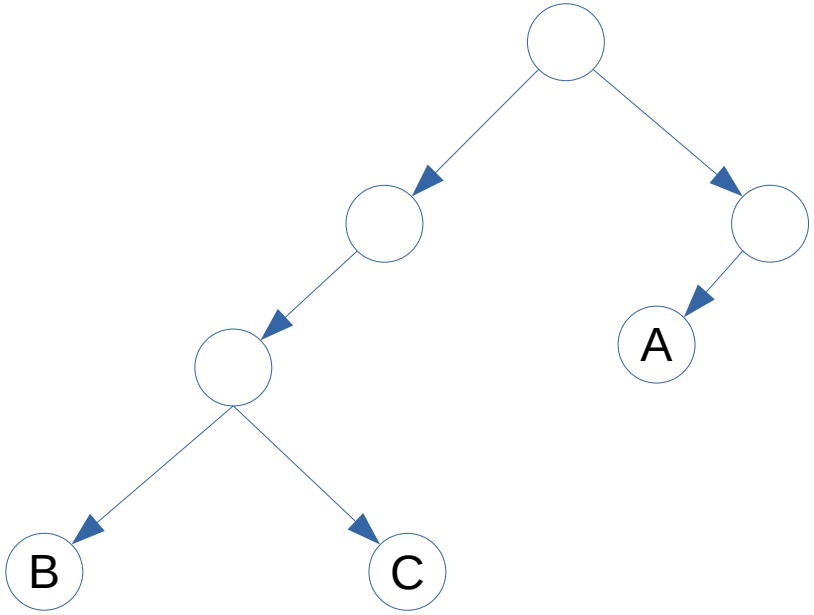
Huffman Coding

Example: Encoding for text

A B C A B

A: 10
B: 000
C: 001



A B C A B

encoded text => 10 000 001 10 000

Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

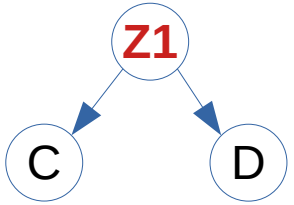
While the container has more than 1 element:

- **Let n1 and n2 be the two elements with least frequency**
- **Create a tree node denoting an "intermediate" element Z that:**
  - **the frequency of "Z" is n1→freq + n2→freq**
  - **the left child of the node is n1→letter**
  - **the right child of the node is n2 →letter**
- **Remove n1 and n2 from the container**
- **Insert Z in the container**

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

Z1

(Z1,2)	(B,2)	(R,2)	(A,5)



Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

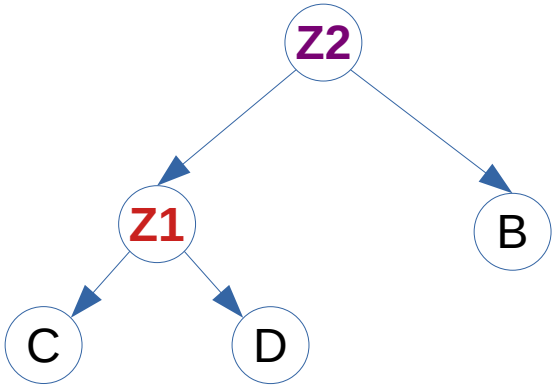
While the container has more than 1 element:

- **Let n1 and n2 be the two elements with least frequency**
- **Create a tree node denoting an "intermediate" element Z that:**
  - **the frequency of "Z" is n1→freq + n2→freq**
  - **the left child of the node is n1→letter**
  - **the right child of the node is n2 →letter**
- **Remove n1 and n2 from the container**
- **Insert Z in the container**

(Z1,2)	(B,2)	(R,2)	(A,5)

Z2

(R,2)	(Z2,4)	(A,5)



Huffman Coding

Create an encoding for text

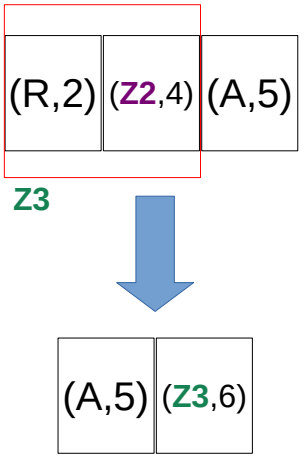
A B R A C A D A B R A

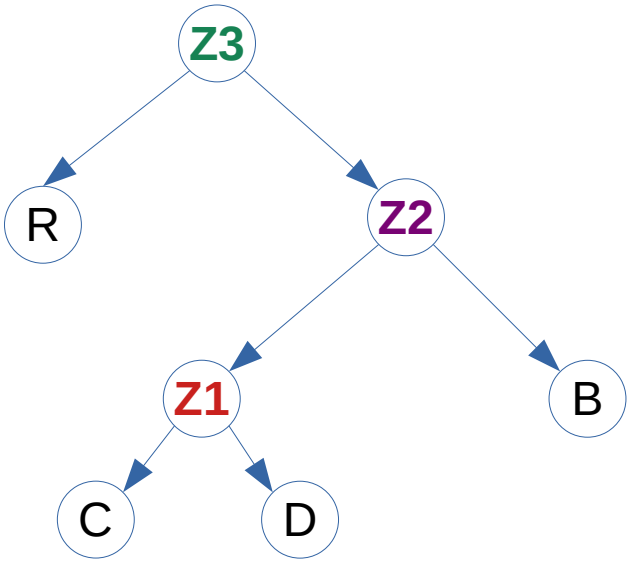
Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

While the container has more than 1 element:

- **Let n1 and n2 be the two elements with least frequency**
- **Create a tree node denoting an "intermediate" element Z that:**
  - **the frequency of "Z" is n1→freq + n2→freq**
  - **the left child of the node is n1→letter**
  - **the right child of the node is n2 →letter**
- **Remove n1 and n2 from the container**
- **Insert Z in the container**





Huffman Coding

Create an encoding for text

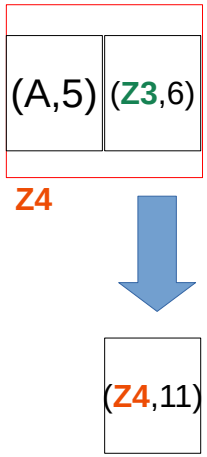
A B R A C A D A B R A

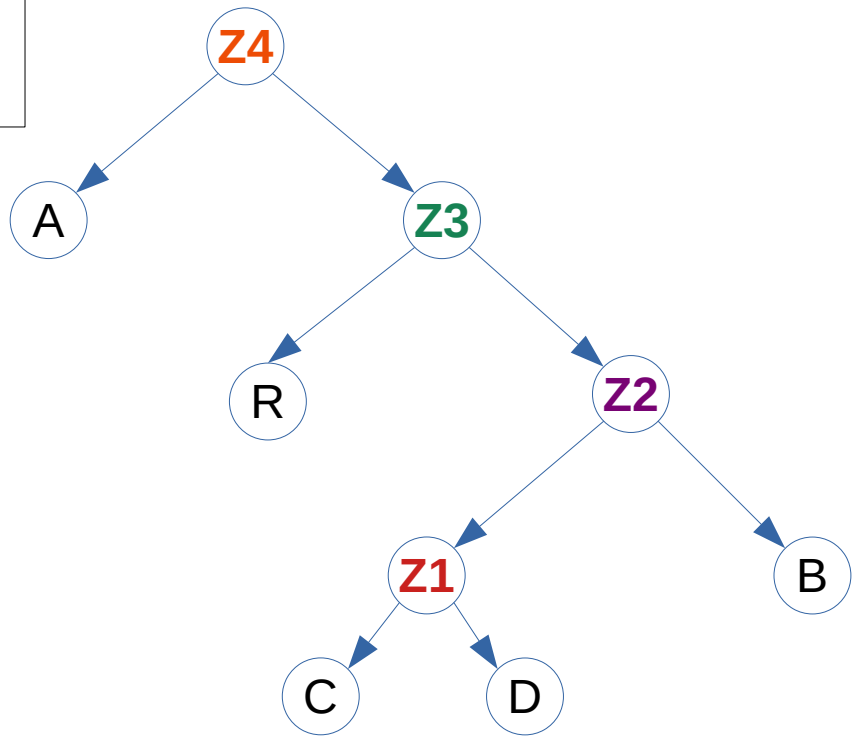
Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

While the container has more than 1 element:

- **Let n1 and n2 be the two elements with least frequency**
- **Create a tree node denoting an "intermediate" element Z that:**
  - **the frequency of "Z" is n1→freq + n2→freq**
  - **the left child of the node is n1→letter**
  - **the right child of the node is n2 →letter**
- **Remove n1 and n2 from the container**
- **Insert Z in the container**

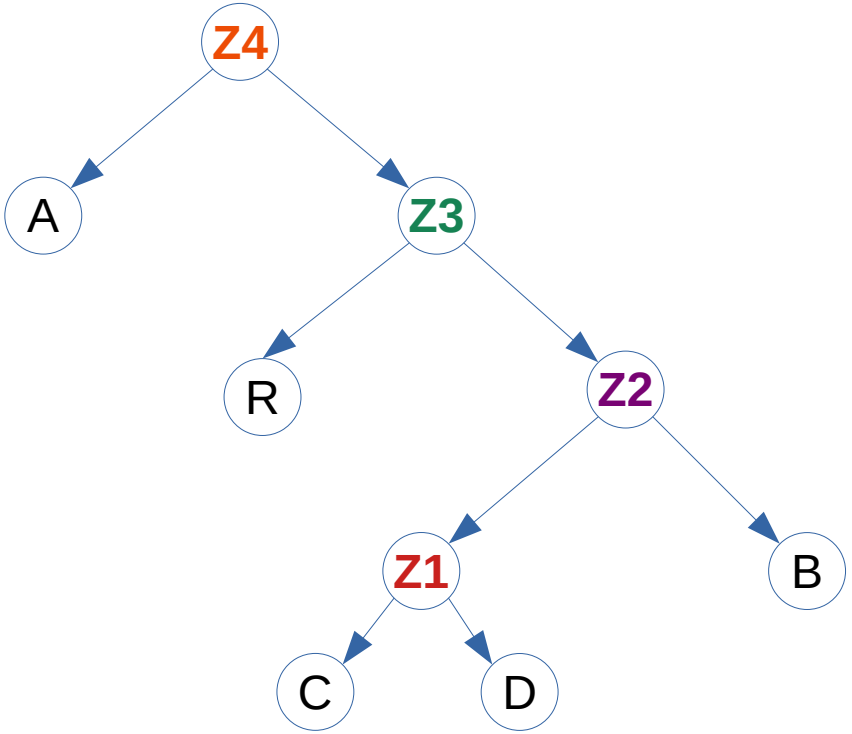




Huffman Coding

Create an encoding for text

A B R A C A D A B R A



A: 0
R: 10
B: 111
C: 1100
D: 1101

Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

While the container has more than 1 element:

- **Let n1 and n2 be the two elements with least frequency**
- **Create a tree node denoting an "intermediate" element Z that:**
  - **the frequency of "Z" is n1→freq + n2→freq**
  - **the left child of the node is n1→letter**
  - **the right child of the node is n2 →letter**
- **Remove n1 and n2 from the container**
- **Insert Z in the container**

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

Z1

(Z1,2)	(B,2)	(R,2)	(A,5)

(C,1) (D,1)

Idea:

- Our elements in the container can be actually nodes of the tree.

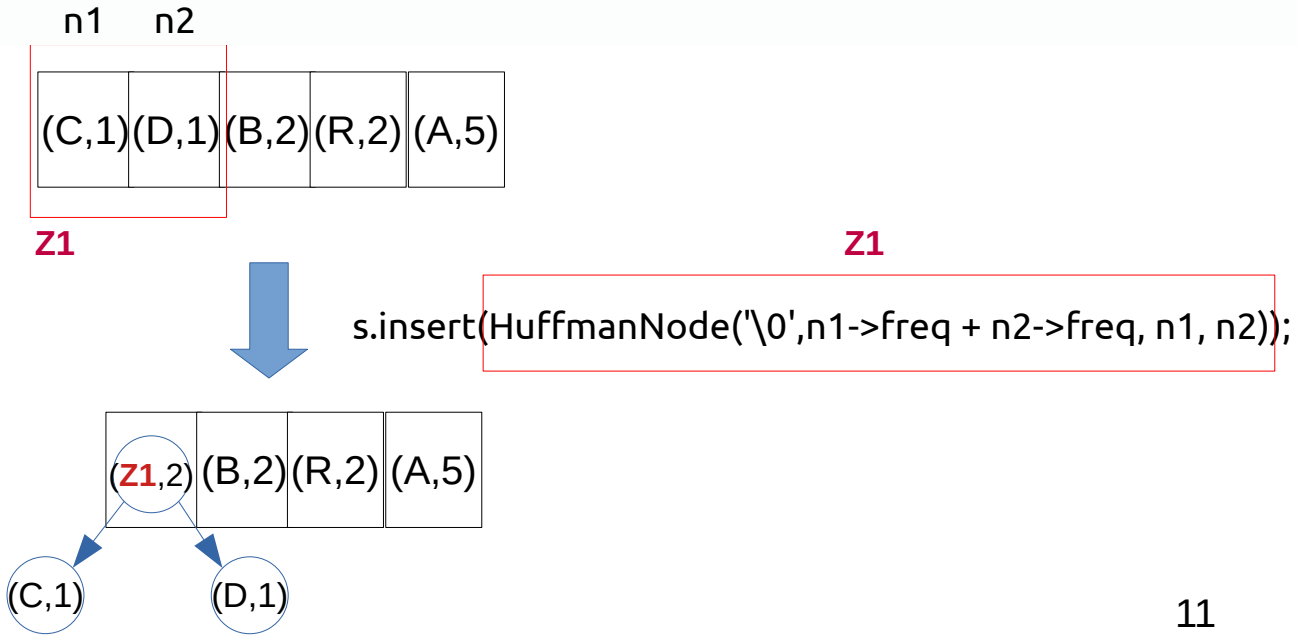
Huffman Coding

Create an encoding for text

Idea:

- Our elements in the container can be actually nodes of the tree.

```
class HuffmanNode
{
    public:

        char letter; // '\0' for non-leaf nodes
        int  freq;

        HuffmanNode* leftChild;
        HuffmanNode* rightChild;

        HuffmanNode(char letter, int freq, HuffmanNode* leftChild = nullptr, HuffmanNode* rightChild = nullptr)
            : letter(letter), freq(freq), leftChild(leftChild), rightChild(rightChild) {}

        HuffmanNode(const HuffmanNode& node)
            : letter(node.letter), freq(node.freq), leftChild(node.leftChild), rightChild(node.rightChild) {}

        bool operator<(const HuffmanNode& node) const
        {
            return this->freq <= node.freq;
        }
};
```



Huffman Coding

Create an encoding for text

A B R A C A D A B R A

Idea:

- Our elements in the container can be actually nodes of the tree.

Create a container where each element x contains a distinct letter of the text and its frequency (keep the contained sorted in ascendant order by the frequency)

(C,1)	(D,1)	(B,2)	(R,2)	(A,5)

```
std::multiset<HuffmanNode> s = { {'D',1}, {'C',1}, {'R',2}, {'B',2}, {'A',5} };
```