Algoritmos e Estruturas de Dados II

Compressão de Dados

Prof. Tiago Eugenio de Melo
tmelo@uea.edu.br

www.tiagodemelo.info

Observações

- As palavras com a fonte `Courier` indicam as palavras-reservadas da linguagem de programação.

Referências

- **Projetos de Algoritmos – com implementações em Pascal e C**. Nivio Ziviani. 2ª edição. Thomson, 2005.

SIGLAS

- RI – Recuperação de Informação.
- NLP – *Natural Language Processing* (Processamento de Linguagem Natural).

INTRODUÇÃO

Compressão de Dados

Compressão de Dados

- Representação de uma fonte de dados de forma mais precisa possível utilizando um menor número de *bits*.

Compressão de Dados

- Representação de uma fonte de dados de forma mais precisa possível utilizando um menor número de *bits*.
- Objetivo é fazer com que a mesma quantidade de informação seja representada em um menor espaço de armazenamento.

Compressão de Dados

- Representação de uma fonte de dados de forma mais precisa possível utilizando um menor número de *bits*.
- Objetivo é fazer com que a mesma quantidade de informação seja representada em um menor espaço de armazenamento.
- Eliminar redundâncias: recorrências de letras, dígitos ou *pixels*.

Compressão de Dados

- Representação de uma fonte de dados de forma mais precisa possível utilizando um menor número de *bits*.
- Objetivo é fazer com que a mesma quantidade de informação seja representada em um menor espaço de armazenamento.
- Eliminar redundâncias: recorrências de letras, dígitos ou *pixels*.
- O receptor deve ser capaz de decodificar os dados para acessar a informação.

Histórico

Histórico

- A compressão de dados é oriunda da Criptografia: algoritmo de compressão é um codificador e um decodificador.

Histórico

- A compressão de dados é oriunda da Criptografia: algoritmo de compressão é um codificador e um decodificador.
- Relatos de encriptação por volta de 1.500 a.C. (escrita cifrada para guardar segredos).

Histórico

- A compressão de dados é oriunda da Criptografia: algoritmo de compressão é um codificador e um decodificador.
- Relatos de encriptação por volta de 1.500 a.C. (escrita cifrada para guardar segredos).
- Gregos e espartanos usavam códigos em movimentos bélicos durante as guerras (475 a.C.).

Histórico

- A compressão de dados é oriunda da Criptografia: algoritmo de compressão é um codificador e um decodificador.
- Relatos de encriptação por volta de 1.500 a.C. (escrita cifrada para guardar segredos).
- Gregos e espartanos usavam códigos em movimentos bélicos durante as guerras (475 a.C.).
- No Século XIX, a invenção do telégrafo e do Código Morse abriu espaço para a criptografia moderna que deixou de ter processos exclusivamente manuais.

Histórico

Histórico

- 1ª Guerra Mundial (1914 a 1918): máquinas de codificação mecânicas usadas para codificar e decodificar textos usando encriptações sofisticadas e complexas.

Histórico

- 1ª Guerra Mundial (1914 a 1918): máquinas de codificação mecânicas usadas para codificar e decodificar textos usando encriptações sofisticadas e complexas.
- 2ª Guerra Mundial (1939 a 1945): codificação da mensagem para esconder a informação que era passada através dos rádios (surge a compactação dos dados).

Histórico

- 1ª Guerra Mundial (1914 a 1918): máquinas de codificação mecânicas usadas para codificar e decodificar textos usando encriptações sofisticadas e complexas.
- 2ª Guerra Mundial (1939 a 1945): codificação da mensagem para esconder a informação que era passada através dos rádios (surge a compactação dos dados).
- Advento dos computadores digitais:

Histórico

- 1ª Guerra Mundial (1914 a 1918): máquinas de codificação mecânicas usadas para codificar e decodificar textos usando encriptações sofisticadas e complexas.
- 2ª Guerra Mundial (1939 a 1945): codificação da mensagem para esconder a informação que era passada através dos rádios (surge a compactação dos dados).
- Advento dos computadores digitais:
  - Necessidade de criação de códigos seguros.

Histórico

- 1ª Guerra Mundial (1914 a 1918): máquinas de codificação mecânicas usadas para codificar e decodificar textos usando encriptações sofisticadas e complexas.
- 2ª Guerra Mundial (1939 a 1945): codificação da mensagem para esconder a informação que era passada através dos rádios (surge a compactação dos dados).
- Advento dos computadores digitais:
  - Necessidade de criação de códigos seguros.
  - Necessidade de redução do espaço de armazenamento.

Motivação

Motivação

- Explosão de informação textual disponível *on-line*:

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.
  - Bancos de dados de documentos.

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.
  - Bancos de dados de documentos.
  - World Wide Web (WWW).

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.
  - Bancos de dados de documentos.
  - World Wide Web (WWW).
- Somente a Web tem hoje 1,714,748,192 de páginas estáticas disponíveis.

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.
  - Bancos de dados de documentos.
  - World Wide Web (WWW).
- Somente a Web tem hoje 1,714,748,192 de páginas estáticas disponíveis.
  - Cada bilhão de páginas ocupando aproximadamente 10 terabytes de texto.

Motivação

- Explosão de informação textual disponível *on-line*:
  - Bibliotecas digitais.
  - Sistemas de automação de escritórios.
  - Bancos de dados de documentos.
  - World Wide Web (WWW).
- Somente a Web tem hoje 1,714,748,192 de páginas estáticas disponíveis.
  - Cada bilhão de páginas ocupando aproximadamente 10 terabytes de texto.
  - Há uma estimativa do Google ter mais de 60 trilhões de páginas indexadas.

Características necessárias para sistemas de RI

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Características necessárias para sistemas de RI

- Métodos de compressão de dados permitem:

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Características necessárias para sistemas de RI

- Métodos de compressão de dados permitem:
  - Pesquisar diretamente o texto comprimido mais rapidamente do que o texto original.

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Características necessárias para sistemas de RI

- Métodos de compressão de dados permitem:
  - Pesquisar diretamente o texto comprimido mais rapidamente do que o texto original.
  - Obter maior compressão em relação a métodos tradicionais, gerando maior economia de espaço.

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Características necessárias para sistemas de RI

- Métodos de compressão de dados permitem:
  - Pesquisar diretamente o texto comprimido mais rapidamente do que o texto original.
  - Obter maior compressão em relação a métodos tradicionais, gerando maior economia de espaço.
  - Acessar diretamente qualquer parte do texto comprimido sem necessidade de descomprimir todo o texto desde o início[1].

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Características necessárias para sistemas de RI

- Métodos de compressão de dados permitem:
  - Pesquisar diretamente o texto comprimido mais rapidamente do que o texto original.
  - Obter maior compressão em relação a métodos tradicionais, gerando maior economia de espaço.
  - Acessar diretamente qualquer parte do texto comprimido sem necessidade de descomprimir todo o texto desde o início[1].
- Compromisso espaço versus tempo.

[1] Silva de Moura, Edleno, et al. "Fast and flexible word searching on compressed text." ACM Transactions on Information Systems (TOIS) 18.2 (2000): 113-139.

Por que usar compressão?

Por que usar compressão?

- Compressão de texto

Por que usar compressão?

- Compressão de texto
  - Maneiras de representar o texto original em menos espaço.

Por que usar compressão?

- Compressão de texto
  - Maneiras de representar o texto original em menos espaço.
  - Substituir os símbolos do texto por outros que possam ser representados usando um número menor de *bits* ou *bytes*.

Por que usar compressão?

- Compressão de texto
  - Maneiras de representar o texto original em menos espaço.
  - Substituir os símbolos do texto por outros que possam ser representados usando um número menor de *bits* ou *bytes*.
- Ganho obtido

Por que usar compressão?

- Compressão de texto
  - Maneiras de representar o texto original em menos espaço.
  - Substituir os símbolos do texto por outros que possam ser representados usando um número menor de *bits* ou *bytes*.
- Ganho obtido
  - O texto comprimido ocupa menos espaço de armazenamento.

Por que usar compressão?

- Compressão de texto
  - Maneiras de representar o texto original em menos espaço.
  - Substituir os símbolos do texto por outros que possam ser representados usando um número menor de *bits* ou *bytes*.
- Ganho obtido
  - O texto comprimido ocupa menos espaço de armazenamento.
  - Menos tempo para ser lido do disco ou ser transmitido por um canal de comunicação e para ser pesquisado.

Por que usar compressão?

Por que usar compressão?

- Preço a pagar

Por que usar compressão?

- Preço a pagar
  - Custo computacional para codificar e decodificar o texto.

Por que usar compressão?

- Preço a pagar
  - Custo computacional para codificar e decodificar o texto.
- Avanço da tecnologia

Por que usar compressão?

- Preço a pagar
  - Custo computacional para codificar e decodificar o texto.
- Avanço da tecnologia
  - Há estimativas de que em 20 anos, o tempo de acesso a discos magnéticos tem se mantido praticamente constante, enquanto a velocidade de processamento aumentou aproximadamente 2 mil vezes.

Por que usar compressão?

- Preço a pagar
  - Custo computacional para codificar e decodificar o texto.
- Avanço da tecnologia
  - Há estimativas de que em 20 anos, o tempo de acesso a discos magnéticos tem se mantido praticamente constante, enquanto a velocidade de processamento aumentou aproximadamente 2 mil vezes.
  - Aumento do investimento no poder de computação em compressão em troca de menos espaço em disco ou menor tempo de transmissão.

Razão de Compressão

Razão de Compressão

- Definida pela porcentagem que o arquivo comprimido representa em relação ao tamanho do arquivo não comprimido.

Razão de Compressão

- Definida pela porcentagem que o arquivo comprimido representa em relação ao tamanho do arquivo não comprimido.
- Exemplo: se o arquivo não comprimido possui 100 *bytes* e o arquivo comprimido resultante possui 30 *bytes*, então a razão de compressão é de 30%.

Razão de Compressão

- Definida pela porcentagem que o arquivo comprimido representa em relação ao tamanho do arquivo não comprimido.
- Exemplo: se o arquivo não comprimido possui 100 *bytes* e o arquivo comprimido resultante possui 30 *bytes*, então a razão de compressão é de 30%.
- Utilizada para medir **o ganho em espaço** obtido por um método de compressão.

Outros importantes aspectos a considerar

Outros importantes aspectos a considerar

- Além da economia de espaço, deve-se considerar:

Outros importantes aspectos a considerar

- Além da economia de espaço, deve-se considerar:
  - Velocidade de compressão e descompressão.

Outros importantes aspectos a considerar

- Além da economia de espaço, deve-se considerar:
  - Velocidade de compressão e descompressão.
  - Possibilidade de realizar casamento de cadeias diretamente do texto comprimido.

Outros importantes aspectos a considerar

- Além da economia de espaço, deve-se considerar:
  - Velocidade de compressão e descompressão.
  - Possibilidade de realizar casamento de cadeias diretamente do texto comprimido.
  - Permitir acesso direto a qualquer parte do texto comprimido e iniciar a descompressão a partir da parte acessada.

Outros importantes aspectos a considerar

- Além da economia de espaço, deve-se considerar:
  - Velocidade de compressão e descompressão.
  - Possibilidade de realizar casamento de cadeias diretamente do texto comprimido.
  - Permitir acesso direto a qualquer parte do texto comprimido e iniciar a descompressão a partir da parte acessada.
- Um sistema de RI para grandes coleções de documentos que estejam comprimidos necessitam acesso direto a qualquer ponto do texto comprimido.

Código de Huffman

O que é?

O que é?

- É um método de compactação que usa as probabilidades de ocorrências dos símbolos no conjunto de dados a ser compactado para determinar códigos de tamanho variável para cada símbolo.

Procedimento

Procedimento

- É construída uma árvore binária baseada na frequência de uso das letras do alfabeto de modo que as mais frequentemente utilizadas apareçam mais perto da raiz.

Procedimento

- É construída uma árvore binária baseada na frequência de uso das letras do alfabeto de modo que as mais frequentemente utilizadas apareçam mais perto da raiz.
- A árvore é construída de baixo para cima (das folhas para a raiz), começando a partir das letras menos usadas até atingir a raiz.

Procedimento

Procedimento

- Etapas:

Procedimento

- Etapas:
  - Cálculo da frequência de cada caractere no arquivo.

Procedimento

- Etapas:
  - Cálculo da frequência de cada caractere no arquivo.
  - Execução do algoritmo de Huffman para a construção de uma árvore binária.

Procedimento

- Etapas:
  - Cálculo da frequência de cada caractere no arquivo.
  - Execução do algoritmo de Huffman para a construção de uma árvore binária.
  - Codificação propriamente dita.

Compressão de textos em linguagem natural

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.
  - Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto.

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.
  - Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto.
  - As implementações tradicionais do método de Huffman consideram caracteres como símbolos.

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.
  - Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto.
  - As implementações tradicionais do método de Huffman consideram caracteres como símbolos.
  - Para aliar as necessidades dos algoritmos de compressão às necessidades dos sistemas de RI apontadas acima, deve-se considerar palavras como símbolos a serem codificados.

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.
  - Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto.
  - As implementações tradicionais do método de Huffman consideram caracteres como símbolos.
  - Para aliar as necessidades dos algoritmos de compressão às necessidades dos sistemas de RI apontadas acima, deve-se considerar palavras como símbolos a serem codificados.
  - Métodos de Huffman baseados em caracteres comprimem o texto para aproximadamente **60%**.

Compressão de textos em linguagem natural

- Um dos métodos de codificação mais conhecidos é o de Huffman (1952):
  - Atribui códigos mais curtos a símbolos com frequências altas.
  - Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto.
  - As implementações tradicionais do método de Huffman consideram caracteres como símbolos.
  - Para aliar as necessidades dos algoritmos de compressão às necessidades dos sistemas de RI apontadas acima, deve-se considerar palavras como símbolos a serem codificados.
  - Métodos de Huffman baseados em caracteres comprimem o texto para aproximadamente **60%**.
  - Métodos de Huffman baseados em palavras comprimem o texto para valores pouco acima de **25%**.

Métodos de Huffman baseados em palavras: vantagens

Métodos de Huffman baseados em palavras: vantagens

- Permitem acesso randômico a palavras dentro do texto comprimido.

Métodos de Huffman baseados em palavras: vantagens

- Permitem acesso randômico a palavras dentro do texto comprimido.
- Considerar palavras como símbolos significa que a tabela de símbolos do codificador é exatamente o vocabulário do texto.

Métodos de Huffman baseados em palavras: vantagens

- Permitem acesso randômico a palavras dentro do texto comprimido.
- Considerar palavras como símbolos significa que a tabela de símbolos do codificador é exatamente o vocabulário do texto.
  - Isso permite uma integração natural entre o método de compressão e o arquivo invertido.

Compressão de Huffman usando palavras

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.
  - Conta suas frequências e gera um código de Huffman para as suas palavras.

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.
  - Conta suas frequências e gera um código de Huffman para as suas palavras.
  - A seguir, comprime o texto substituindo cada palavra pelo seu código.

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.
  - Conta suas frequências e gera um código de Huffman para as suas palavras.
  - A seguir, comprime o texto substituindo cada palavra pelo seu código.
- Assim, a compressão é realizada em duas passadas sobre o texto:

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.
  - Conta suas frequências e gera um código de Huffman para as suas palavras.
  - A seguir, comprime o texto substituindo cada palavra pelo seu código.
- Assim, a compressão é realizada em duas passadas sobre o texto:
  - Obtenção da frequência de cada palavra diferente.

Compressão de Huffman usando palavras

- Técnica de compressão mais eficaz para textos em linguagem natural.
- O método considera cada palavra diferente do texto como um símbolo.
  - Conta suas frequências e gera um código de Huffman para as suas palavras.
  - A seguir, comprime o texto substituindo cada palavra pelo seu código.
- Assim, a compressão é realizada em duas passadas sobre o texto:
  - Obtenção da frequência de cada palavra diferente.
  - Realização da compressão.

Forma eficiente de lidar com palavras e separadores

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:
  - Espaço, vírgula, ponto, ponto e vírgula, interrogação, entre outros.

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:
  - Espaço, vírgula, ponto, ponto e vírgula, interrogação, entre outros.
- Uma forma eficiente de lidar com palavras e separadores é representar o espaço simples de forma implícita no texto comprimido.

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:
  - Espaço, vírgula, ponto, ponto e vírgula, interrogação, entre outros.
- Uma forma eficiente de lidar com palavras e separadores é representar o espaço simples de forma implícita no texto comprimido.
  - Nesse modelo, se uma palavra é seguida de um espaço, então somente a palavra é codificada.

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:
  - Espaço, vírgula, ponto, ponto e vírgula, interrogação, entre outros.
- Uma forma eficiente de lidar com palavras e separadores é representar o espaço simples de forma implícita no texto comprimido.
  - Nesse modelo, se uma palavra é seguida de um espaço, então somente a palavra é codificada.
  - Senão, a palavra e o separador são codificados separadamente.

Forma eficiente de lidar com palavras e separadores

- Um texto em linguagem natural é constituído de palavras e separadores.
- Separadores são caracteres que aparecem entre palavras:
  - Espaço, vírgula, ponto, ponto e vírgula, interrogação, entre outros.
- Uma forma eficiente de lidar com palavras e separadores é representar o espaço simples de forma implícita no texto comprimido.
  - Nesse modelo, se uma palavra é seguida de um espaço, então somente a palavra é codificada.
  - Senão, a palavra e o separador são codificados separadamente.
- No momento da decodificação, supõe-se que um espaço simples segue cada palavra, a não ser que o próximo símbolo corresponda a um separador.

Árvore de Huffman

Árvore de Huffman

- O método de Huffman produz a árvore de codificação que minimiza o comprimento do arquivo compactado.

Árvore de Huffman

- O método de Huffman produz a árvore de codificação que minimiza o comprimento do arquivo compactado.
- Existem diversas árvores que produzem a mesma compressão:

Árvore de Huffman

- O método de Huffman produz a árvore de codificação que minimiza o comprimento do arquivo compactado.
- Existem diversas árvores que produzem a mesma compressão:
  - Por exemplo, trocar o filho à esquerda de um nó por um filho à direita leva a uma árvore de codificação alternativa com a mesma razão de compressão.

Árvore de Huffman

- O método de Huffman produz a árvore de codificação que minimiza o comprimento do arquivo compactado.
- Existem diversas árvores que produzem a mesma compressão:
  - Por exemplo, trocar o filho à esquerda de um nó por um filho à direita leva a uma árvore de codificação alternativa com a mesma razão de compressão.
  - Entretanto, a escolha preferencial para a maioria das aplicações é a **árvore canônica**.

Árvore de Huffman

- O método de Huffman produz a árvore de codificação que minimiza o comprimento do arquivo compactado.
- Existem diversas árvores que produzem a mesma compressão:
  - Por exemplo, trocar o filho à esquerda de um nó por um filho à direita leva a uma árvore de codificação alternativa com a mesma razão de compressão.
  - Entretanto, a escolha preferencial para a maioria das aplicações é a **árvore canônica**.
    - Uma árvore de Huffman é canônica quando a altura da subárvore à direita de qualquer nó nunca é menor que a altura da subárvore à esquerda.

Árvore de Huffman

Árvore de Huffman

- A representação do código na forma de árvore facilita a visualização.

Árvore de Huffman

- A representação do código na forma de árvore facilita a visualização.
- Sugere métodos de codificação e decodificação triviais:

Árvore de Huffman

- A representação do código na forma de árvore facilita a visualização.
- Sugere métodos de codificação e decodificação triviais:
  - Codificação: a árvore é percorrida emitindo *bits* ao longo de suas arestas.

Árvore de Huffman

- A representação do código na forma de árvore facilita a visualização.
- Sugere métodos de codificação e decodificação triviais:
  - Codificação: a árvore é percorrida emitindo *bits* ao longo de suas arestas.
  - Decodificação: os *bits* de entrada são usados para selecionar as arestas.

Árvore de Huffman

- A representação do código na forma de árvore facilita a visualização.
- Sugere métodos de codificação e decodificação triviais:
  - Codificação: a árvore é percorrida emitindo *bits* ao longo de suas arestas.
  - Decodificação: os *bits* de entrada são usados para selecionar as arestas.
  - Essa abordagem é ineficiente tanto em termos de espaço quanto em termos de tempo.

Algoritmo baseado na codificação canônica com comportamento linear em tempo e espaço

Algoritmo baseado na codificação canônica com comportamento linear em tempo e espaço

- O algoritmo é atribuído a Moffat e Katajainen (1995)

Algoritmo baseado na codificação canônica com comportamento linear em tempo e espaço

- O algoritmo é atribuído a Moffat e Katajainen (1995)
- Calcula os comprimentos dos códigos em lugar dos códigos propriamente ditos.

Algoritmo baseado na codificação canônica com comportamento linear em tempo e espaço

- O algoritmo é atribuído a Moffat e Katajainen (1995)
- Calcula os comprimentos dos códigos em lugar dos códigos propriamente ditos.
- A compressão atingida é a mesma, independentemente dos códigos utilizados.

Algoritmo baseado na codificação canônica com comportamento linear em tempo e espaço

- O algoritmo é atribuído a Moffat e Katajainen (1995)
- Calcula os comprimentos dos códigos em lugar dos códigos propriamente ditos.
- A compressão atingida é a mesma, independentemente dos códigos utilizados.
- Após o cálculo dos comprimentos, há uma forma elegante e eficiente para a codificação e a decodificação.

Exemplo

Exemplo

- Considere a seguinte sequência de caracteres:

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF
- A menor codificação possível para os caracteres é em binário de três bits por caractere.

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF
- A menor codificação possível para os caracteres é em binário de três bits por caractere.
- Portanto:

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF

- A menor codificação possível para os caracteres é em binário de três bits por caractere.

Caractere	A	B	C	D	E	F
Código	000	001	010	011	100	101

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF

- A menor codificação possível para os caracteres é em binário de três bits por caractere.
- Portanto:

Caractere	A	B	C	D	E	F
Código	000	001	010	011	100	101

- Gerando:

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF

- A menor codificação possível para os caracteres é em binário de três bits por caractere.
- Portanto:

Caractere	A	B	C	D	E	F
Código	000	001	010	011	100	101

- Gerando:

  000000000000000000001001001001001010010010010011011011100100101

Exemplo

- Considere a seguinte sequência de caracteres:

  AAAAAABBBBBBCCCCDDDEEF

- A menor codificação possível para os caracteres é em binário de três bits por caractere.
- Portanto:

Caractere	A	B	C	D	E	F
Código	000	001	010	011	100	101

- Gerando:

000000000000000000001001001001001010010010010011011011100100101

63 bits de comprimento

Exemplo (cont.)

Exemplo (cont.)

- Para usar o código de Huffman, é necessário montar uma árvore.

Exemplo (cont.)

- Para usar o código de Huffman, é necessário montar uma árvore.
- O primeiro passo é contar as ocorrências de cada símbolo da cadeia:

Exemplo (cont.)

- Para usar o código de Huffman, é necessário montar uma árvore.
- O primeiro passo é contar as ocorrências de cada símbolo da cadeia:

Caractere	A	B	C	D	E	F
Contagem	6	5	4	3	2	1

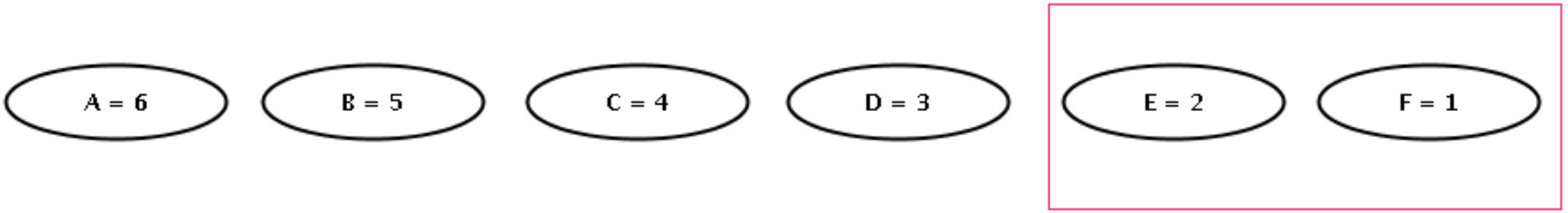
Exemplo (cont.)

Exemplo (cont.)

- Passo 1

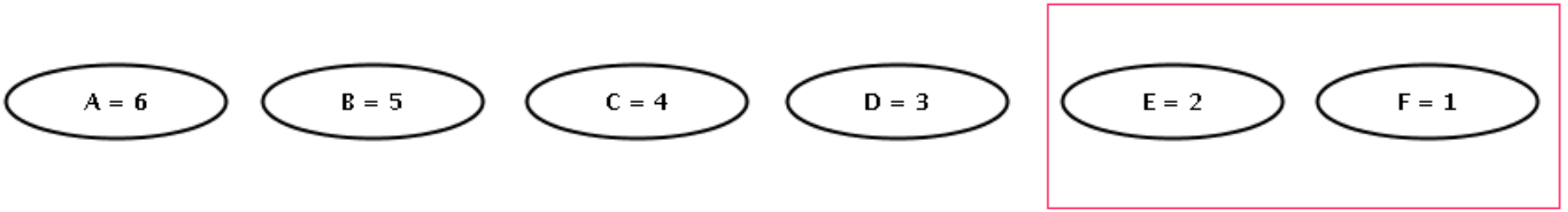
Exemplo (cont.)

- Passo 1



Exemplo (cont.)

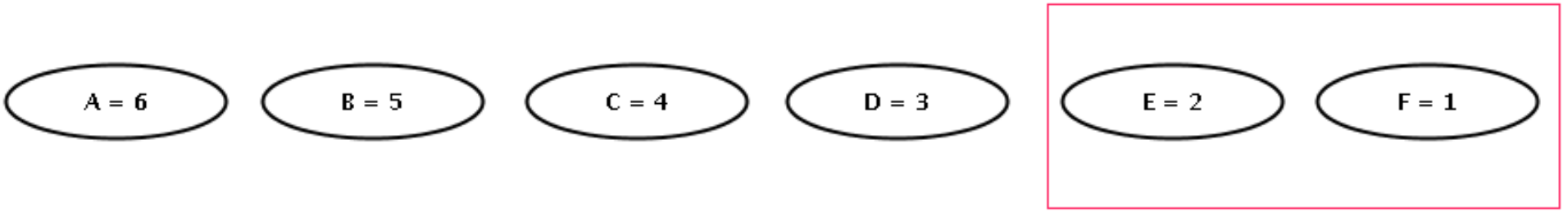
- Passo 1



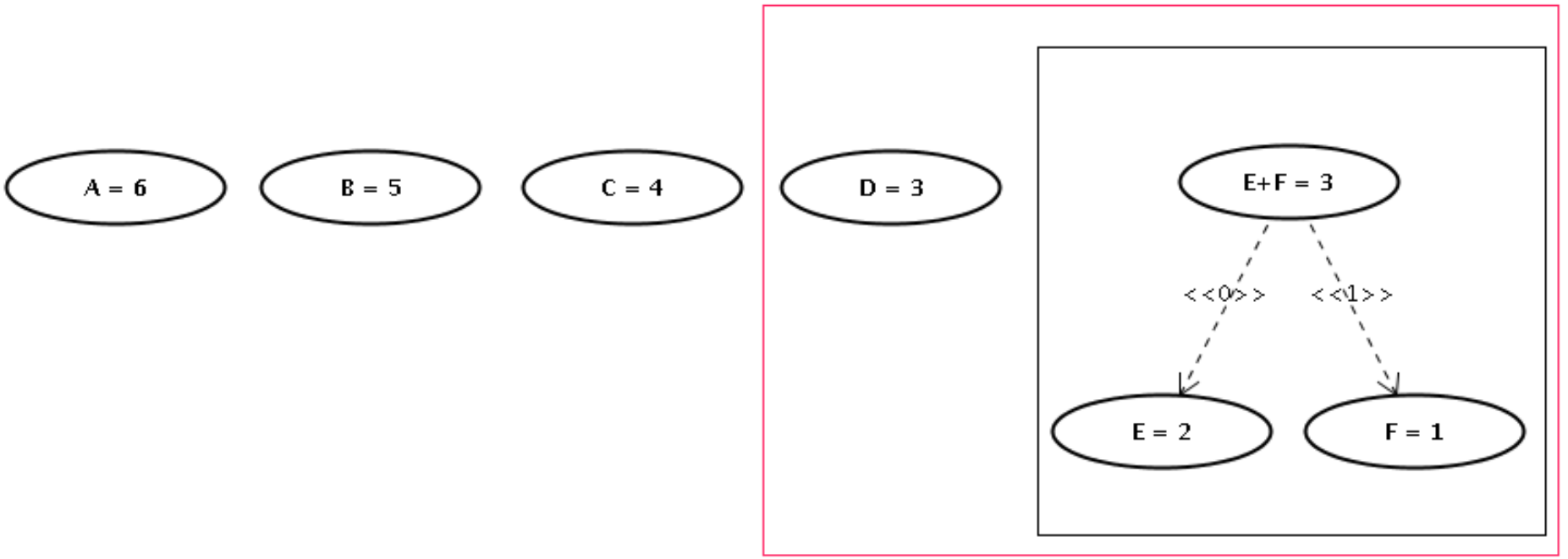
- Passo 2

Exemplo (cont.)

- Passo 1



- Passo 2



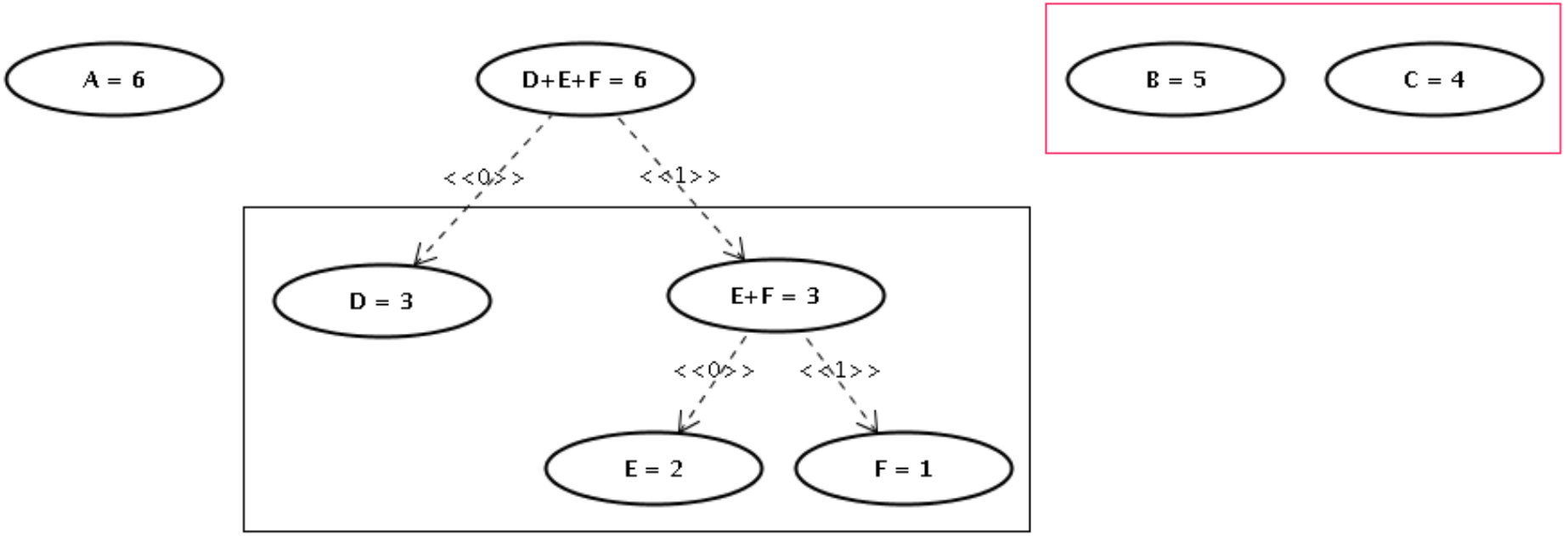
Exemplo (cont.)

Exemplo (cont.)

- Passo 3

Exemplo (cont.)

- Passo 3



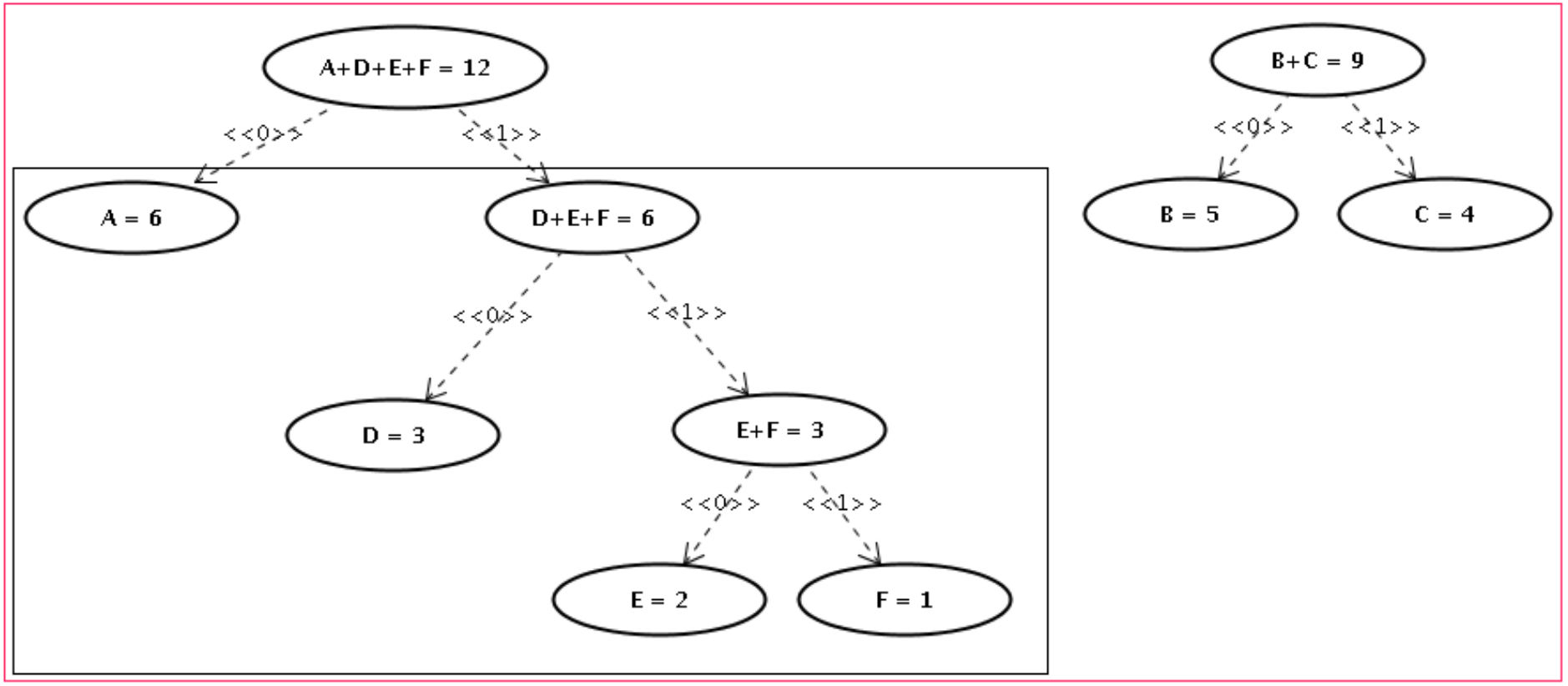
Exemplo (cont.)

Exemplo (cont.)

- Passo 4

Exemplo (cont.)

- Passo 4



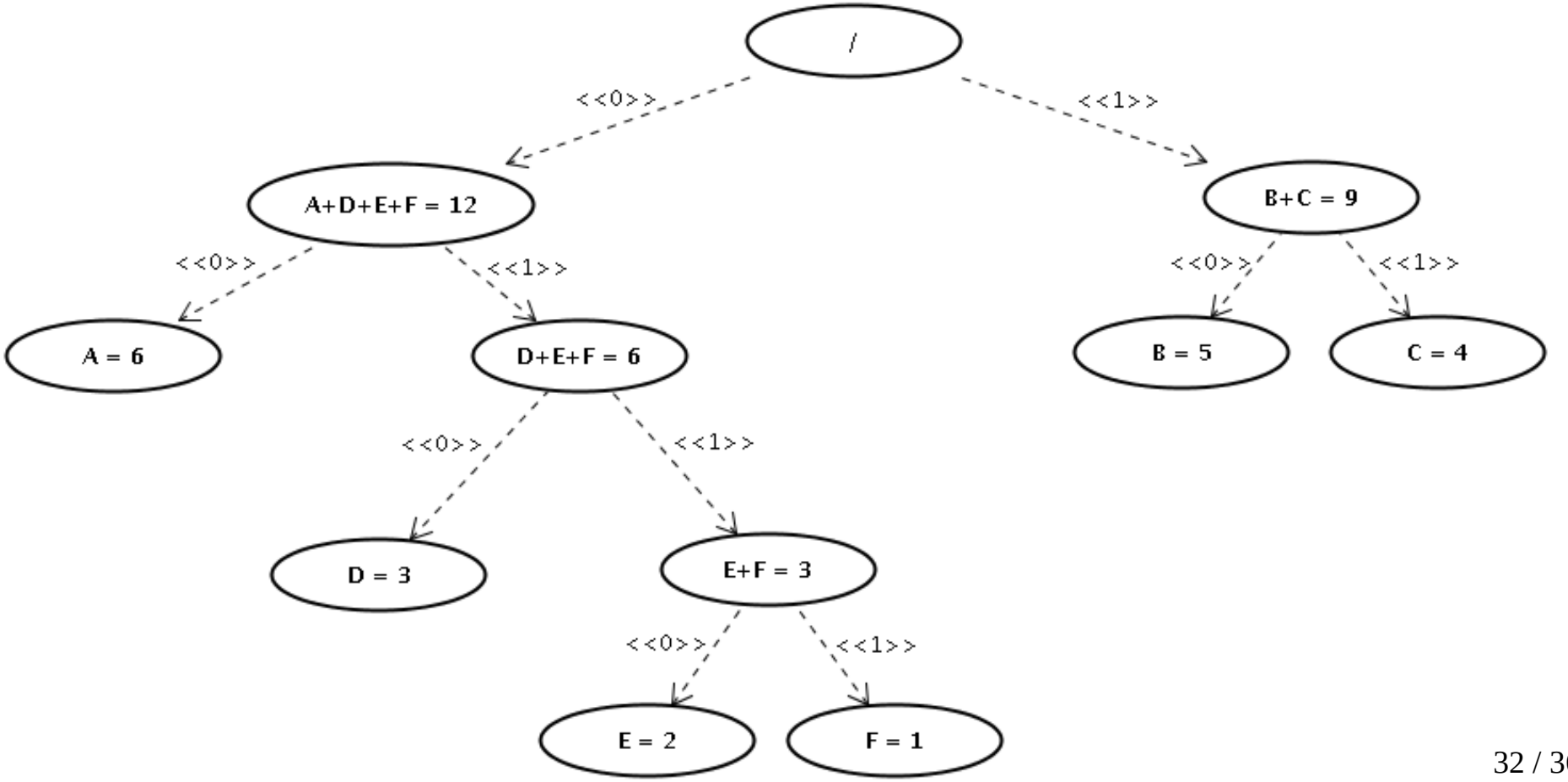
Exemplo (cont.)

Exemplo (cont.)

- Árvore de Huffman

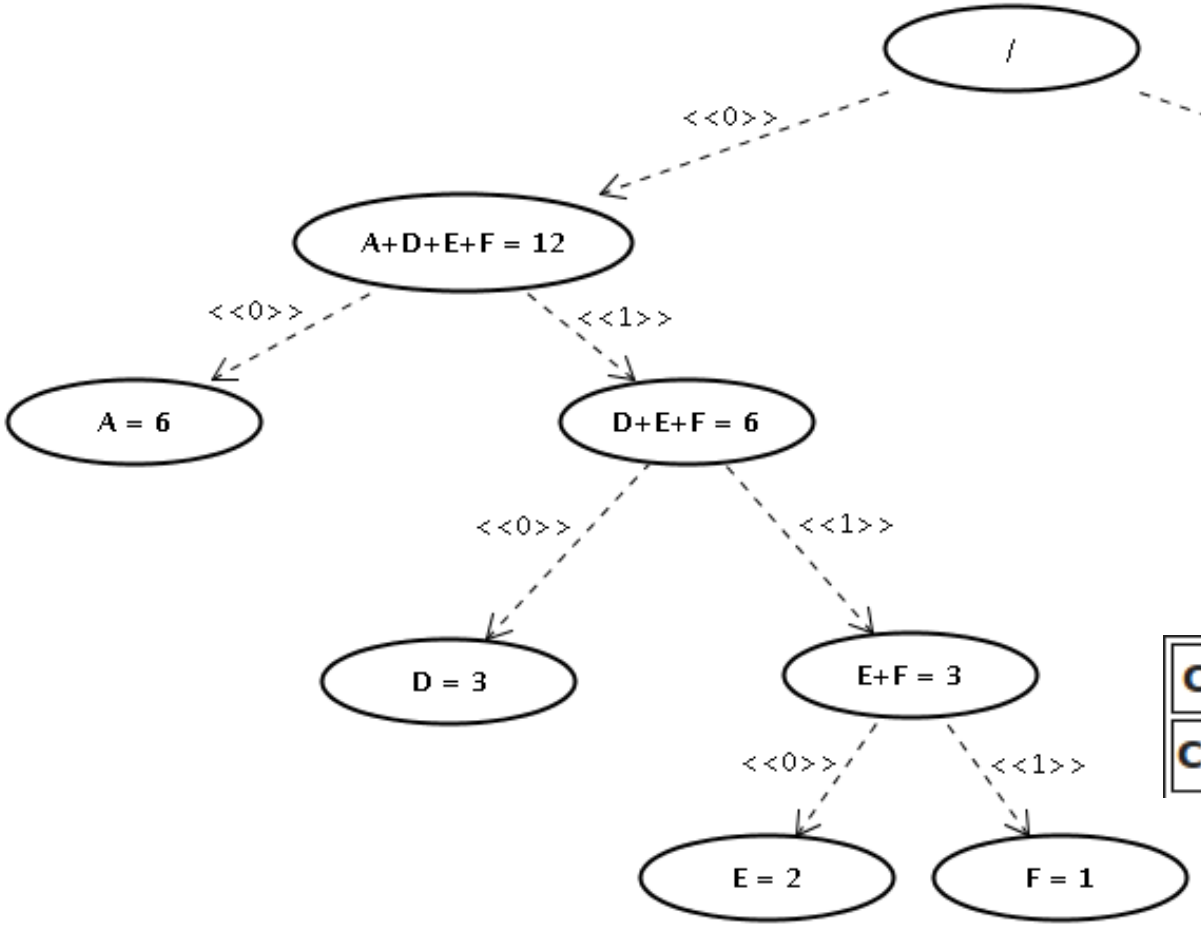
Exemplo (cont.)

- Árvore de Huffman



Exemplo (cont.)

- Árvore de Huffman



Caractere	A	B	C	D	E	F
Contagem	00	10	11	010	0110	0111

Exemplo (cont.)

Exemplo (cont.)

- A sequência original agora é representada por:

Exemplo (cont.)

- A sequência original agora é representada por:
- 000000000000101010101011111111010010010011001100111

Exemplo (cont.)

- A sequência original agora é representada por:
- 000000000001010101010111111110100100100110011100111

Caractere	A	B	C	D	E	F
Contagem	00	10	11	010	0110	0111

Exemplo (cont.)

- A sequência original agora é representada por:
- 000000000001010101010111111110100100100110011100111
- Totalizando 51 bits.

Caractere	A	B	C	D	E	F
Contagem	00	10	11	010	0110	0111

Exemplo (cont.)

- A sequência original agora é representada por:
- 000000000001010101010111111110100100100110011100111
- Totalizando 51 bits.
- A compressão foi de 12 bits ou cerca de 20%.

Caractere	A	B	C	D	E	F
Contagem	00	10	11	010	0110	0111

Exercícios

Exercícios

- Desenhar a árvore de Huffman para o seguinte conjunto de chaves e frequências descrito abaixo:
  - {A:1, B:6, C:2, D:1, E:1, F:9, G:2, H:3}
- A árvore fornecida pelo algoritmo de Huffman é única, ou seja, a árvore sempre fornecerá a mesma árvore?