Data compression

Created by anhtt-fit@mail.hut.edu.vn
Updated by huonglt-fit@mail.hut.edu.vn

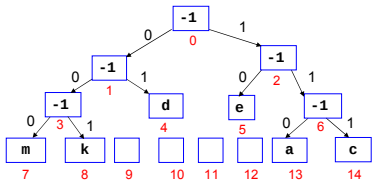
File Compression

- Có thể dùng mã Huffman Code để nén file.
- File nén có thể tổ chức như sau

HM	Huffman Tree	Size	Data …

  - HM: mã bắt đầu file nén
  - Huffman Tree: mảng biểu diễn cây Huffman được sinh từ dữ liệu cần mã hóa
  - Size: kích thước dữ liệu nén, tính theo bits
  - Data: lưu dữ liệu nén dưới dạng các bit

Biểu diễn dạng mảng của cây Huffmam



Các nút lá trên cây Huffman được gán nhãn dưới dạng các ký tự

15	-1	-1	-1	-1	d	e	-1	m	k					a	c

Kích thước mảng = ($2^0 + 2^1 + \ldots + 2^n$) với n là độ sâu của cây
Con của nút i ở tại vị trí 2i+1 và 2i+2
Cha của nút i ở tại vị trí (i-1)/2

Implementation

- Mảng lưu trữ cây Huffman

```
typedef struct {
   int size;
   int * nodes;
} HuffmanTreeArray;
```

- Viết hàm chuyển đổi 1 cây Huffman về dạng mảng:
  - HuffmanTreeArray tree2array(HuffmanTree);

Quiz 1

- Viết lại các hàm trong bài trước để nén các file sử dụng mã Huffman.
- Chương trình cần dùng để nén file, gọi dưới dạng dòng lệnh là:
  - $ compress in_file [out_file]
- Cần cài đặt các hàm sau
  - HuffmanTree makeHuffman (FILE * in);
  - void createHuffmanTable(HuffmanTree htree, Coding* htable);
  - HuffmanTreeArray tree2array(HuffmanTree);
  - void compressFile(FILE* in, FILE *out);

Giải nén file

- Trước tiên, kiểm tra tiền tố "HM" của file
- Đọc cây trong mảng
- Khi nhận được cây, quét xâu bit nhận được
  - Dữ liệu ở dạng bit chứ không phải dạng byte
- Thuật toán quét
  - Đặt con trỏ ở gốc của cây
  - Nếu nút hiện tại có giá trị -1, đọc bit mới
    - 0 ⇒ chuyển con trỏ đến cây con trái
    - 1 ⇒ chuyển con trỏ đến cây con phải
  - Ngược lại, nhận ký tự mới tại nút, chuyển con trỏ đến gốc

Quiz 2

- Viết chương trình giải nén file được nén trong Quiz 1
- Sử dụng chế độ dòng lệnh để giải nén file:
  - $ decompress compressed_file [out_file]