CSE100 Discussion 6

Wenjia Ouyang
Announcement

• 1. PA3
  – Additional pre-grade: on Thursday 8pm
  – Final deadline: November 12 @ 8:00pm
  – 2 points extra credit if your implementation beats the reference.

• 2. Section C: Midterm
  – Check your score @gradesource
  – Mean: 84.5/100
PA 3 Implementation strategy

• Step 1. Create HC Tree: HCNode and HCTree files

**HCNode.cpp:**

- bool comp(HCNode* one, HCNode* other)
- bool HCNode::operator<(const HCNode& other)

**HCTree.cpp:**

- build()
  - Write verification code to check that you can construct simple Huffman trees correctly

- encode(), decode()
  - Test with simple inputs
• Step 2. Add binary I/O: `BitInputStream/BitOutputStream`
  – `readBit`, `readByte`, `readInt`
  – `writeBit`, `writeByte`, `writeInt`
Buffering and bit-by-bit I/O

• Basic idea: use a byte as an 8-bit buffer
  – Use bitwise shift and or operators to write individual bits into the byte, or read individual bits from it;
  – flush the byte when it is full, or done with I/O
/** Read the next bit from the bit buffer.  
 * Fill the buffer from the input stream first if needed.  
 * Return 1 if the bit read is 1;  
 * return 0 if the bit read is 0.  
 */  

int readBit() {  
  // If all bits in the buffer are read, fill the buffer first  
  // Get the bit at the appropriate location in the bit  
  // buffer, and return the appropriate int  
  // Increment the index  
}
/** Write the least significant bit of the argument to 
* the bit buffer, and increment the bit buffer index. 
* But flush the buffer first, if it is full. 
*/

void writeBit(int i)  {
    // Is the bit buffer full? Then flush it

    // Write the least significant bit of i into the buffer
    // at the current index

    // Increment the index
}
• Step 3: compress/uncompress

  ```
  ./compress myInputFile myOutputFile(compressed file)
  ```

• In the reference solution, the header is just 256 ints in a row, one for each byte value.

• This header takes 1024 bytes of space. (4*256)

• How could you **beat** it?
  – Hint: how to create the header?
• **Step 3: compress/uncompress**
  
  – Check is the given input/output file valid?
  
  – Compress:
    
    • Read char from input file and build the freq vector
    
    • Build Huffman tree:
      – myHCTree.build(myFreqs)
    
    • Write header to output file
    
    • Encode:
      – myHCTree.encode(myByte, myOutputFile)
  
  – Uncompress
    
    • Read header from input file
    
    • Build Huffman tree
    
    • decode
Error Handling

• Does the input file exist?
• Can it be opened and read from?
• Does the file contain any data?
• Does the output file has the same name as the input file?
• Can the output file be written?

• Be familiar with istream, ostream, ifstream, ofstream, and their functions.
Section C: Midterm

• Red-Black Tree
  A RBT is a BST where:
  1. All nodes are red or black
  2. Root is always black
  3. Red nodes have only black children
  4. All paths from root through null pointers have the same number of black nodes
Q5. d) Design a “worst-case” red-black tree with 10 nodes, i.e., a red-black tree with the longest possible path from the root to a leaf.
Questions?

Good luck on your PA3 and HAPPY HALLOWEEN!