



Microprogetto “Alberi di Huffman permanenti”

1. Rappresentazione “permanente” di alberi di Huffman

Completa la definizione della classe `HuffmanTree` per gestire compattamente le operazioni di salvataggio e ripristino relative ad alberi di Huffman. Più specificamente, il protocollo deve comprendere i seguenti costruttori e metodi:

```
public HuffmanTree( int[] frqs )   costruttore: creazione dell'albero di Huffman  
                                   a partire dall'array frqs delle frequenze dei byte  
  
public HuffmanTree( String src )   costruttore: creazione (ripristino) dell'albero di Huffman  
                                   a partire dalla sua codifica testuale contenuta nel file di nome src  
  
public Node root()                 restituisce la radice dell'albero di Huffman rappresentato dall'oggetto  
  
public void save( String dst )     salva l'albero di Huffman (in forma testuale) in un file di nome dst
```

I file riferiti attraverso i parametri `src` e `dst` contengono la codifica linearizzata di un albero di Huffman. In altri termini è come se questi file fossero la versione dell'albero di Huffman resa *permanente* nel file system e separata da uno specifico contesto di utilizzo. In particolare, il file “sorgente” il cui nome è passato come argomento al secondo costruttore deve contenere solo la codifica di un albero di Huffman; simmetricamente, il file “destinazione” argomento del metodo `save` viene creato dal metodo stesso e alla fine conterrà solo la codifica dell'albero di Huffman inglobata dall'oggetto di tipo `HuffmanTree`. Per quanto riguarda il resto, l'argomento del primo costruttore è un array delle frequenze dei byte in base alle quali viene costruito l'albero di Huffman seguendo le modalità consuete; il metodo `root`, infine, restituisce la radice, e quindi l'intera struttura, dell'albero rappresentato dall'oggetto di tipo `HuffmanTree`. Ai fini della realizzazione della classe `HuffmanTree` si può naturalmente utilizzare, invocandoli direttamente, i metodi statici definiti nei “moduli” `HuffmanCompression` e `HuffmanDecompression`.

2. Verifica

Per concludere, verifica sperimentalmente le funzionalità della nuova classe applicando i seguenti frammenti di codice (uno alla volta, utilizzando le funzionalità di *BlueJ*, oppure racchiusi in un nuovo metodo ad hoc):

```
int[] frqs = Huffman.charHistogram( "Node.java" );  
HuffmanTree tree1 = new HuffmanTree( frqs );  
tree1.save( "H1.txt" );  
  
HuffmanTree tree2 = new HuffmanTree( "H1.txt" );  
tree2.save( "H2.txt" );
```

Il contenuto dei file `H1.txt` e `H2.txt`, che può essere letto attraverso un editor di testo, deve essere lo stesso.