



## Homework #4

**Due Date: 11:59pm Monday 14 April 2014**

In this homework you will write a C++ program that can compress and decompress images using dictionary coding. Your program will read the image on `stdin` and outputs the resulting image on `stdout`, and get in the required operation on the command line.

Please present a report containing your answers as well as a zip file containing all your code.

1. [3.5 points] Implement the function `encode()` to encode an image using the LZ77 algorithm and output the total number of bits required on the first line, followed by the triplets  $\langle j, k, x \rangle$  one per line. The triplets represent the match position  $j$ , the match length  $k$ , and the code of the symbol following the match i.e. three integers separated by a space. Try the function on the small images `msg1.pgm`, `msg2.pgm` and `msg3.pgm` to make sure it's working correctly.

Note: we will only use *fixed-length* coding for the triplets, so the number of bits required depend on the size of the search buffer  $S$ , the size of the lookahead buffer  $T$ , and the size of the alphabet.

2. [2.5 points] Implement the function `decode()` to decode an image encoded using the LZ77 algorithm and output the corresponding PPM image. Note that you will need to pass in the image metadata i.e. the number of rows, columns, and maximum value on the command line, since these are not included in the encoded message. Try the function on the output of encoding the small images `msg1.pgm`, `msg2.pgm` and `msg3.pgm` to make sure it's working correctly.
3. [3 points] Try your encoder on the four 256x256 images shown in class (`sen`, `sensin`, `earth`, `omaha`) and compute the compression ratio and number of bits required when using different values for the parameters  $S$  and  $T$ .

Using the best parameter value from above, try encoding the *difference* values instead of the pixel values. In particular, for every pixel, instead of encoding the pixel value, encode the difference between that pixel and the pixel on the left (except the first column which are encoded as is).

Do you get similar numbers to the numbers given in the book? Comment on your answer.

## Command Line

You need to modify the main file `hw04.cpp` to include the required functionality. Your program should be named `hw04`, and should be called as follows:

- To encode an input image and print out the total number of bits required on the first line followed by the triplets, one per line:

```
./hw04 -encode S T A < input.ppm
```

where  $S$  is length of the search buffer,  $T$  is the size of the lookahead buffer, and  $A$  is the number of symbols in the alphabet, the input image is called `input.ppm` and the output is written to `stdout`. For example, to encode `msg1.pgm` using a search buffer of 256, a lookahead buffer of 16, an alphabet of 3 symbols, you could run:

```
./hw04 -encode 256 16 3 < msg1.pgm
```

- To decode an image encoded above:

```
./hw04 -decode xres yres max < image.enc
```

where the encoded image is called `image.enc` and the original image had  $xres$  columns,  $yres$  rows, and maximum intensity  $max$ . For example, to decode the encoded version of `sena.pgm`, using a search buffer of 1024, a lookahead buffer of 256, and 256 symbols in the alphabet, you could run:

```
./hw04 -encode 1024 256 256 < sena.pgm | ./hw04 -decode 256 256 255
```

## **Instructions**

- All code should be implemented in C++ under Linux.
- Please submit your homework in one zip file named as follows: *CMPN206.HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *CMPN206.HW01.Mohamed.Aly.zip*.
- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

## **Grading**

- 9 points: requirements above
- 1 point: submission instructions