



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

SCC0222 – Laboratório de Introdução à Ciência da Computação I

Trabalho 03: Run-Length

Professor: Dr. Rodrigo Fernandes de Mello (mello@icmc.usp.br)
Estagiário PAE: Fábio Henrique Gomes Sikansi (fhenrique@usp.br)
Martha Dais Ferreira (daismf@icmc.usp.br)
Monitor: Fernando Candiani (fncandiani@usp.br)
Lucas Parras (parraslucas@gmail.com)
Loys Gibertoni (loys.gibertoni@usp.br)
Colaborador: Felipe Simões Lage Gomes Duarte (fgduarte@icmc.usp.br)

1 Objetivo do Trabalho

Você deverá implementar o algoritmo run-length para a compactação e descompactação de uma imagem no formato PGM.

2 Imagem PGM

A designação de formato de imagem PBM (Portable Bitmap) engloba três formatos de imagem para imagens a preto e branco, em escala de tons cinzentos e a cores, todos eles sem compressão e que apresentam uma estrutura comum. Estes três tipos de formato de imagens são:

- PBM (Portable BitMap) - imagens a preto e branco (sem tons de cinzento)
- PGM (Portable GrayMap) - imagens em tons de cinzento
- PPM (Portable PixMap) - imagens a cores

A definição original destes formatos teve em vista permitir a transmissão de imagens por meio de correio eletrônico que até então não permitia a transmissão de ficheiros anexados, binários ou não. A definição do formato foi mais tarde modificada para permitir a representação binária dos conteúdos das imagens.

Os formatos de imagem PGM são constituídos pelos seguintes campos:

```
<magic number>
<Altura da imagem 'm'> <Largura da imagem 'n'>
<Valor máximo dos tons de cinza>
<valor do pixel (0,0)> ... <valor do pixel (0,n)>
.
.
.
.
.
<valor do pixel (m,0)> ... <valor do pixel (m,n)>
```

O Identificador do tipo de formato (designado por “magic number”), é determinado de acordo com Tipo, ASCII ou Binário

Tipo	ASCII	Binário
PBM	P1	P4
PGM	P2	P5
PPM	P3	P6

Em particular, no nosso caso, iremos trabalhar com imagens PGM do tipo ASCII, ou seja, “P2”. Assim, um exemplo de imagem PGM que o programa deve ser capaz de processar é:

```

P2
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

3 Run-Length

Run-length (ou RLE) é uma técnica para comprimir cadeias de caracteres onde existem sequências longas de caracteres repetidos. O princípio do funcionamento dessa codificação é simples: Quando temos a ocorrência de uma repetição contínua de determinado caractere, por exemplo, BAAAAA-AAAAAAC, é possível substituir sua representação pelo par (A, 12) precedido por um identificador resultado em uma string do tipo 'B @ A 12 C' (assumido neste exemplo que o identificador é o @).

Na compressão de imagens esta técnica é mais promissora pois imagens apresentam maiores áreas contínuas de uma mesma cor. Desenhos e outras imagens com número limitados de cores tendem a gerar melhores resultados usando esta técnica.

4 Proposta

O sistema deve ser capaz de compactar uma imagem **pgm** do tipo P2 e descompactar uma imagem compactada pelo RLE que aqui denotaremos como tipo P8.

4.1 Compactação

A compactação consiste em ler os dados da imagem no formato PGM-P2 (como exemplificado acima) e gerar um output no **stdout** com a imagem já compactada. O exemplo abaixo é o resultado da compactação do exemplo dado anteriormente:

```

P8
24 7
15
@ 0 24
0 @ 3 4 0 0 @ 7 4 0 0 @ 11 4 0 0 @ 15 4 0
0 3 @ 0 5 7 @ 0 5 11 @ 0 5 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 @ 15 4 0
0 3 @ 0 5 7 @ 0 5 11 @ 0 5 15 @ 0 4
0 3 @ 0 5 @ 7 4 0 0 @ 11 4 0 0 15 @ 0 4
@ 0 24

```

Observe que a substituição do número pela tríade (demarcador, cor do pixel e frequência) só é vantajoso quando o número de vezes que um mesmo pixel aparece em sequência é maior que 3, ou seja, em um cenário que o número de um mesmo pixel apareça 3 vezes ou menos, a saída deverá ser o próprio número e não a representação reduzida. Por exemplo, em uma sequência

'1 2 2 3 3 3 4 4 4 4 5 5 5 6 6 6 6'

o resultado final deverá ser

'1 2 2 3 3 3 @ 4 4 5 5 5 @ 6 4'

Atente para o fato que no fim de cada linha não existe um espaço em branco, o caractere de quebra de linha deve preceder imediatamente o ultimo elemento da linha.

4.2 Descompactação

A descompactação consiste em ler os dados da imagem compactada no formato RLE-P8 (como exemplificado acima) e gerar um output no **stdout** com a imagem PGM-P2. Novamente, atente para

o fato que no fim de cada linha não existe um espaço em branco, o caractere de quebra de linha deve preceder imediatamente o ultimo elemento da linha.