Universidade do Minho
MIETI, 20919-2020
Tecnologias e Serviços Multimédia

Trabalho Prático N°1

A. Programa em C ou C++ que mediante um ficheiro de entrada, gere uma tabela de codificação tal como a que foi obtida por codificação Shannon-Fano. Os símbolos a considerar são bytes individuais. O TP é de realização individual e o código deve ser entregue e testado na aula prática de 24 de Fevereiro. Não é necessário ser entregue qualquer relatório.

   Passo 1 - Calcular as probabilidades dos símbolos.
   Passo 2 - Cálculo dos códigos da tabela.
   Passo 3 - Assimilar: símbolo -> código do símbolo -> probabilidade.

   Exemplo:

   'ABBCDAAA'

   P(A) = 4/8
   P(B) = 2/8
   P(C) = 1/8
   P(D) = 1/8

SIMBOLO	PROBABILIDADE	CODIGO BINARIO
A	4/8	0
B	2/8	10
C	1/8	110
D	1/8	111

   [:tempo de execução]

B. Continuação do programa do ponto A para realizar a codificação do ficheiro de entrada num ficheiro comprimido de saída com o resultado de aplicar a tabela de códigos calculada ao ficheiro de entrada. O TP é de realização individual e o código deve ser entregue até final do dia 7 de Março e testado na aula prática de 8 de Março, se requerido pelo docente. Não é necessário ser entregue qualquer relatório.

   Exemplo do conteúdo do ficheiro de saída do exemplo do ponto anterior:

   (4,4,2,1,1,8)[01010110][111000XX]

   Em que,

   (I,I,I,I,I,I) – seis valores inteiros
   [bbbbbbbb] – um byte em binário

Os primeiros 6 valores são um exemplo do que pode ser um "header" do ficheiro de compressão com a informação necessária para depois se fazer a descompressão. Neste caso, o primeiro valor indica o número de símbolos no ficheiro, os quatro valores seguintes são a frequência de cada um dos símbolos e o sexto valor é o número de símbolos codificados. Outros tipos de cabeçalhos podem ser utilizados e vários tipos de optimizações podem ser implementados (ou planeados), conforme discutido nas aulas.

C. Implementação dum programa de descompressão/descodificação que execute a operação inversa à implementada no programa dos passos anteriores. Esta fase pode gerar um programa novo ou uma nova versão do anterior que detete automaticamente se o ficheiro é para comprimir/descomprimir. O output deste programa deve gerar um ficheiro igual ao ficheiro original antes de ser comprimido. Durante a execução, o programa deve calcular a tabela de Shannon-Fano a partir dos dados do "header" e descodificar a "stream" binária comprimida. O TP é de realização individual e o código deve ser entregue até final do dia 15 de Março e testado na aula prática de 16 de Março, se requerido pelo docente. Não é obrigatório a entrega de qualquer relatório adicional.

D. Na última fase do trabalho os alunos devem tentar implementar alguns mecanismos de otimização que achem relevantes. Ficam aqui algumas sugestões:

   i. Devido a possíveis problemas de concentração espacial ou temporal de símbolos, os métodos de codificação/compressão estatísticos como o método Shannon-Fano perdem eficácia. Este problema pode ser atacado com a utilização dum mecanismo simples de compressão por sequências de símbolos, ou *Run-Length Encoding* (RLE). Existem variadíssimas variantes deste mecanismo e deve ser utilizado antes da aplicação da codificação estatística. Pode ser aplicado tendo como referência um alfabeto de símbolos diferente do usado no mecanismo de codificação estatístico (muitas vezes aplica-se o RLE tendo em conta que os símbolos são os bits "0" e "1").

   ii. Outra forma de lutar contra a concentração espacial ou temporal de símbolos é realizar a codificação por blocos independentes, i.e., divide-se o ficheiro (ou uma sequência temporal de símbolos da fonte de informação) em blocos do mesmo tamanho e aplica-se o método de codificação estatístico a cada bloco, ou seja, calcula-se uma tabela estatística para cada bloco e gera-se um "header" e uma sequência de codificação para cada bloco. Este método necessita que os blocos tenham uma dimensão razoável. Para ficheiros pequenos, o mecanismo adota um único bloco. Cada bloco deve ser lido do ficheiro para memória RAM, deve ser analisado e codificado numa única passagem e a sequência codificada deve ser armazenada em memória RAM e, só no final do processamento do bloco inteiro é que deve ser escrita em ficheiro de saída.

iii. Na compressão, de forma a acelerar o processamento, podem usar-se tabelas estatísticas dinâmicas, i.e., à medida que a sequência de símbolos dum bloco vai sendo tratada a tabela estatística usada no mecanismo de codificação estatístico vai sendo atualizada/melhorada, sendo que a tabela começa com o mesmo valor de probabilidade (ou de número de ocorrências) para todos os símbolos. Essa tabela inicial é usada durante um período de tempo (ou número de símbolos a codificar) pequeno ao fim do qual a tabela de códigos é atualizada com as probabilidades/ocorrências atualizadas. À medida que o tempo vai passando (ou a sequência de símbolos vai sendo processada) a qualidade estatística da tabela vai melhorando. Com esta otimização não é necessário escrever o número de ocorrências de cada símbolo em "headers" pois a informação das tabelas estatísticas também pode ser calculada diretamente na descompressão, onde se utilizará o mesmo mecanismo de atualização das tabelas estatísticas. Obviamente, este mecanismo é menos eficaz na codificação e descodificação (ou seja, o tamanho médio dos códigos é superior) quando comparado com o método tradicional de tabelas não-dinâmicas, mas é muito mais rápido a executar.

iv. Na descompressão, para mais rapidamente se identificar o símbolo equivalente à sequência de bits dos códigos que vão sendo lidos, pode utilizar-se uma procura em árvores binárias, em que a tabela estatística é representada através duma árvore binária onde os valores dos símbolos estão armazenados nas folhas da árvore. Este método pode ser utilizado também quando são usadas tabelas dinâmicas. Nesse caso, a árvore de identificação é atualizada sempre que a tabela de códigos for atualizada.

v. A codificação por blocos é mais eficiente em termos estatísticos e da sua aplicação podem resultar melhores valores de compressão, ainda que a sua execução seja mais lenta pois implica a criação e manipulação de tabelas estatísticas muito maiores. Os alunos podem criar uma opção de execução em que seja aplicada uma codificação por blocos de dois símbolos, i.e., cada símbolo a considerar será composto por dois bytes (16 bits) em vez dum único byte (8 bits).

As ferramentas criadas devem ser testadas com ficheiros de teste de vários tamanhos cujos *links* serão fornecidos pelo docente.

As ferramentas deverão ter um modo de funcionamento em *debug* (ou *verbose*) em que imprimam informações relevantes durante a sua execução (como por exemplo, tamanho dos blocos de leitura, dos blocos de processamento estatístico, do valor da entropia (dinâmica ou fixa), das tabelas de códigos, dos valores do comprimento médio dos códigos, do valor de compressão, dos tempos de execução

parciais/totais, etc.)

Os alunos devem documentar/explicar o código criado através de comentários no próprio código e através dum pequeno relatório. Este documento deve descrever:

- As estratégias escolhidas, as opções tomadas e os mecanismos adotados, incluindo eventuais otimizações;
- Os valores de desempenho da execução das ferramentas (valor da compressão obtida e tempo de execução) quando aplicadas aos ficheiros de teste indicados pelo docente;
- As principais limitações das ferramentas desenvolvidas e a forma como se poderiam ultrapassar.

O TP é de realização individual e o código final e o relatório deve ser entregue até final do dia 29 de Março e discutido e defendido na aula prática de 30 de Março, se requerido pelo docente.