Relatório EPSub

Ísis Ardisson Logullo – 7577410

Nesse EpSub utilizei algumas classes auxiliares de SW: http://algs4.cs.princeton.edu/code Para a compressão, utilizei uma fila de prioridade para construir a trie necessária para o algoritmo de Huffman, a partir do vetor de frequências. Monto uma tabela de símbolos com os códigos necessários para essa compressão. Ela é a tabela de códigos que leva cada caractere (8 bits) no seu código. As chaves são os caracteres e os valores os códigos. Essa tabela é uma trie binária que será convertida na correspondente cadeia de bits. Com essa tabela produzo os bits codificados. Uso o stream de saida para gravar o arquivo bin com essa codificação do algoritmo.

O stream de saida vê o tipo de entrada para salvar no arquivo de saida correspondente. Se a entrada for um txt, a saida é uma compressão, logo terá um arquivo bin. Se for um arquivo bin de entrada, logo será uma decodificação e a saida será um txt. Na decodificação cada código de caracter é convertido no correspondente caractere. A tabela de códigos tem códigos diferentes para caracteres diferentes e nenhum código é prefixo de outro. Uso uma busca por cada uma das n chaves e essas chaves estão nas folhas da trie

O stream de entrada recebe o nome do arquivo de entrada para extrair o texto. Ele é tratado para ler binários. Tudo é setado na classe correspondente de leitura de entradas. Por se tratar de binários foi necessário uma classe especial para entrada e saida. Foram testados arquivos curtos e grandes. Os 3 casos de teste seguem em anexo. O makefile pode ajudar apenas trocando o nome do arquivo de entrada.

Teste1: txt – 12bytes; bin convertido – 15bytes; txt convertido – 12bytes
Teste2: txt – 2,1kb; bin convertido – 1,3kb; txt convertido – 2,1kb
Teste3: txt – 4,4mb; bin convertido – 2,5mb; txt convertido – 4,4mb

Com isso percebe-se que um arquivo muito pequeno nâo compensa ser compactado, pois fica maior que o original. Conforme vai aumentando ele vai diminuindo em torno da metade do tamanho original.