

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 saída para gravar o arquivo bin com essa codificação do algoritmo.

Esse stream de saída vê o tipo de entrada para salvar no arquivo de saída correspondente. Se a entrada for um txt, a saída é uma compressão, logo terá um arquivo bin. Se for um arquivo bin de entrada, logo será uma decodificação e a saída será um txt. Na decodificação cada código de caractere 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

Tive alguns problemas para pegar as infos do arquivo quando é um bin de entrada. Utilizo o Scanner do Java para pegar todos os dados dos arquivos e passar para os métodos. Vejo a extensão do arquivo de entrada e por ela aciono o método certo. Um arquivo txt irá acionar a compactação e um arquivo bin irá acionar a descompactação.