mp3\_20181594 알고리즘 과제

20181594 구산

input.txt에서 문자열을 입력받는다.

입력받은 문자열에서 가장 적게 사용된 문자 순서대로 정렬하고, 트리를 만든다.

제일 빈도수가 작은 빈도수를 가진 노드와 두번째로 작은 빈도수를 가진 노드끼리 합하여, 부모 노드는 두 빈도수를 합한 빈도수를 갖도록 한다. 가장 빈도수가 작은 노드는 왼쪽, 두번째로 작은 노드는 오른쪽 자식이 된다. 그렇게 모든 노드를 합친 뒤에, 가장 top 노드부터 탐색을 진행하는데, 왼쪽 노드로 가면 문자열에 0을 추가하고, 오른쪽 노드로 가면 문자열에 1을 추가한다. 그렇게 각 문자에 대해 이진수로 이루어진 문자열을 할당하고, 각 문자가 나올때마다 그 이진수 문자열을 ‘encoding’ 문자열에 추가한다. 각 문자의 빈도수는 fprintf로 출력하여 decoding시 트리를 만들기 위한 준비를 해놓는다.

이렇게 만들어진 이진수 배열을, 8개 단위의 비트로 묶어서 0~255의 확장된 아스키 코드 문자 하나에 대응시켜서, 출력한다. 마지막 8개 미만의 비트의 경우 따로 계산하여 따로 표기한다.

그렇게 input.txt.zz가 만들어지면, 위에서부터 2개의 int형 자료들을 읽어들여 len, -1의 자료가 나올때까지 읽는다. -1을 감지하면, break하고 그 자료들을 이용해 트리를 만든다.

텍스트 파일에 있는 아스키 코드들을 다시 이진수 배열로 확장하고, 그 배열에서 트리를 이용해 압축된 파일을 해독해서 출력하면 된다.

압축률은 주로 45% 내외의 압축률을 보였다. 원래 한 문자당 8비트이고, 평균적으로 각 문자당 3~4비트의 숫자만으로 표현이 가능했으므로, 합당한 수치이다.

압축 전 / 후

1000 / 433 byte 43.3%

2000 / 966 byte 48.3%

10000 / 4196 byte 41.96%

100000 / 42357 byte 42.357%

나타나는 문자가 더 다양하고, 고를수록, 압축률이 떨어지는 모습을 보였다.