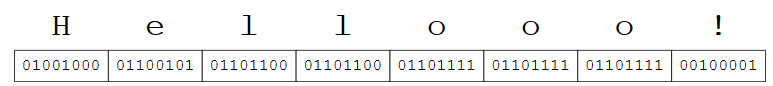
***Ví dụ xây dựng cây nhị phân cho chữ “Hellooo!”***

*Biểu diễn dưới dạng mã ASCII*



*Nén bằng mã 3 – bit coding*

Mỗi ký tự sử dụng 8 bit để biểu diễn nên tổng cộng sẽ dùng 64 bit. Bảng mã ASCII mở rộng dùng 8 bit để biểu diễn 256 ký tự khác nhau. Trong khi đó thông điệp của chúng ta chỉ gồm 5 ký tự khác nhau, như vậy thực tế chỉ cần dùng 3 bit là đủ để phân biệt được 5 ký tự này. Bây giờ ta sẽ liệt kê 5 ký tự phân biệt và gán cho nó một mã nhị phân 3 bit phân biệt. Lúc này ta vẫn có thể biểu diễn trọn vẹn thông điệp đã nén và dùng bảng giải mã này để khôi phục thông điệp ban đầu.

|  |  |
| --- | --- |
| *Nén bằng 3 – bit coding* | *3 bit – coding* |
|  |  |

Như vậy, chỉ cần dùng 3×8=24 bit để biểu diễn thông điệp trên. Mỗi ký tự đều dùng đúng một số lượng bit để biểu diễn, chúng ta gọi cách này là fixed-length encoding (dùng cùng một số lượng bit để mã hóa, các giá trị mã hóa đều khác nhau). Bảng các giá trị được mã hóa cũng là prefix code (là một tập các giá trị mã hóa mà không có phần tử nào được bắt đầu bằng một phần tử khác).

Ta lại nhận thấy rằng, có những phần tử xuất hiện rất nhiều lần, nếu gắn cho chúng một mã có độ dài thấp nhất, còn các phần tử xuất ít hơn có thể có mã dài hơn, thì vẫn có thể tiết kiệm được hơn nữa.

*Tối ưu hóa mã prefix code*

|  |  |
| --- | --- |
| *Giả sử mã prefix code* | *Variable – length encoding* |
|  |  |

Với mã prefix code giải sử ở trên, rõ ràng chỉ cần dùng 18 bit để biểu diễn thông điệp trên. Cách gán mã dựa trên tần suất xuất hiện là ý tưởng của thuật toán Huffman coding.