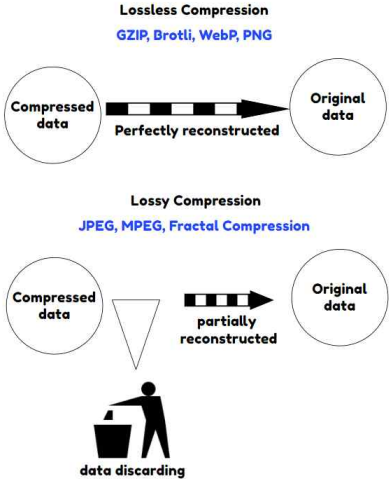
Data Structure / Algorithm - Huffman Code

자료구조와 알고리즘을 왜 배워야 하는 걸까?

Data Structure / Algorithm

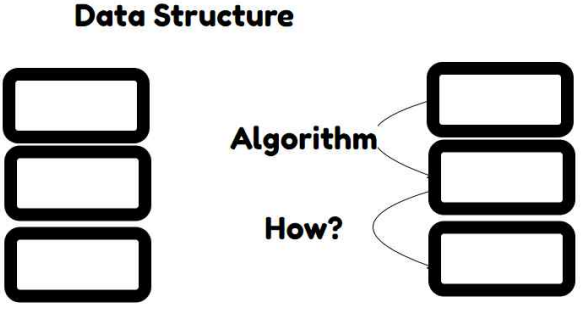
손실 압축 vs 무손실 압축

size가 많이 줄어도 명확한 차이가 없어보임



Why you know DataStructure & Algorithm?

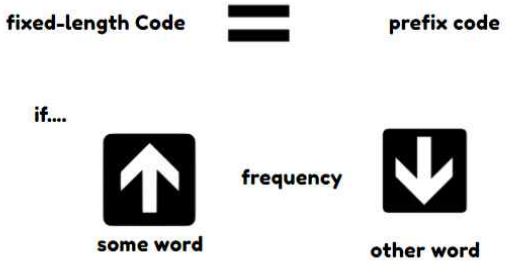
자료구조와 알고리즘을 왜 배워야 하는 걸까?



O(n) more Effective!

fixed-length code vs variable-length code

- 고정 길이 코드 => 일부 단어가 다른 단어보다 전송될 가능성이 더

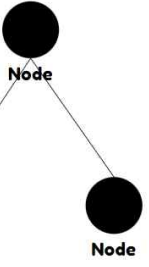


Prefix Code

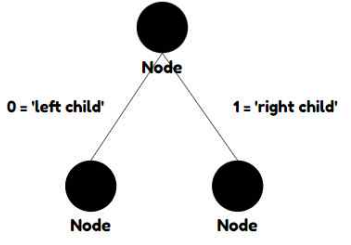
Huffman Code

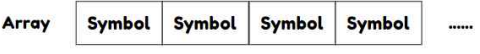
- Huffman Code - lossless 데이터 압축 알고리즘
- prefix code를 만드는 가장 알려진 방식
- prefix Code?
  - prefix 특성으로 구별되는 코드 시스템
  - 시스템 안에 어떤 다른 codeword의 prefix가 다른 단어로 포함되어 있으면 X
  - 즉, 고유하게 해독 가능한 코드
  - 이러한 경우, 특별한 표시 없이 각 단어를 식별 가능

Huffman Code



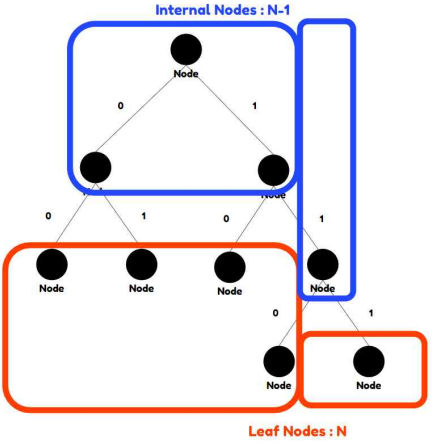
Symbol, Weight, Link





Huffman Code

사용하지 않는 기호를 생략하는
허프만 트리
=> 가장 최적의 코드 길이 생성



Huffman Code

Compression

A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED

이 문자열을 허프만 코드 방식으로 압축한 값을 알려주세요

Huffman Code

A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED

1. 각 문자들의 빈도 순 만들고 정렬

2. leaf Node로 시작 (기호의 확률을 포함하는)

3. 확률이 가장 낮은 두 개의 노드를 선택하고 &
이 두 노드를 자식으로 갖는 새로운 내부 노드 생성

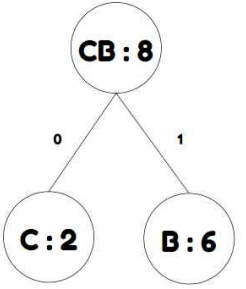
4. 새 노드의 가중치는 자식 가중치의 합으로 설정
(Huffman Tree의 루트인 1개의 노드만 남을 때까지 3,4반복)

Huffman Code

A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED

1.

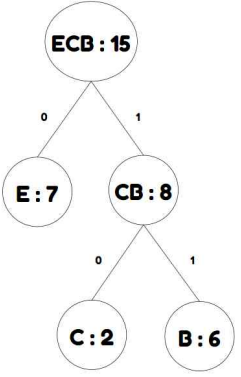
C : 2
B : 6
E : 7
_ : 10
D : 10
A : 11



Huffman Code

2.

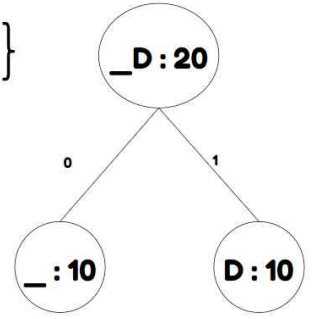
E : 7
CB : 8
_ : 10
D : 10
A : 11



Huffman Code

3.

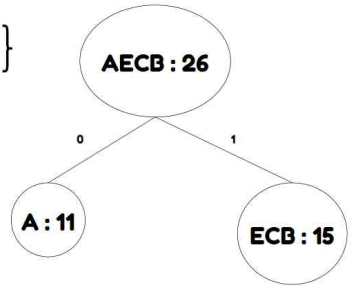
_ : 10
D : 10
A : 11
ECB : 15



Huffman Code

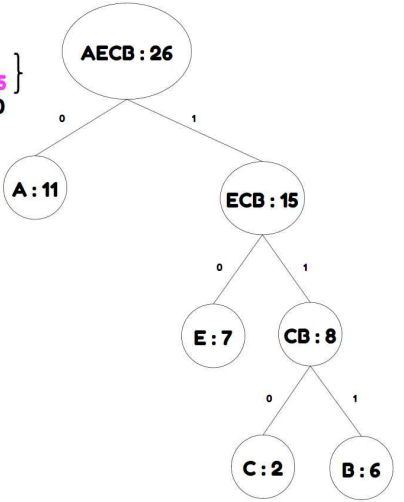
4.

A : 11
ECB : 15
_D : 20

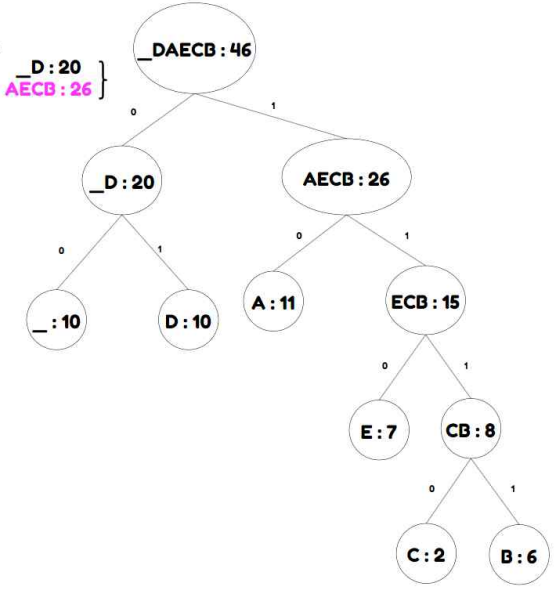


4.

A : 11
ECB : 15
_D : 20



5.



Huffman Coding

_ : 00
D : 01
A : 10
E : 110
C : 1110
B : 1111

_ : 00
D : 01
A : 10
E : 110
C : 1110
B : 1111

Huffman Code

go Compression

A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED

10/00/01/110/10/01/00/01/10/01/00/1110/110/01/110/01/00/10/00/1111/10/
01/00/1111/10/1111/110/00/10/00/1111/110/10/01/110/01/00/10/1111/10/1110/10/00/1111/110/01

console.log("10/00/01/110/10/01/00/01/10/01/00/1110/110/01/110/01/00/10/00/1111/10/01/00/1111/10/1111/110/00/10/00/1111/110/10/01/110/01/00/10/1111/10/1110/10/00/1111/110/01".replaceAll('/','') === "${Your Answer}"); => true가 나오면 됨

Huffman Code

BOJ_6800 풀어보는 것도 추천!

기본 개념 익히는 데는 충분한 듯

참고

Huffman Coding

https://en.wikipedia.org/wiki/Huffman_coding

https://en.wikipedia.org/wiki/Prefix_code