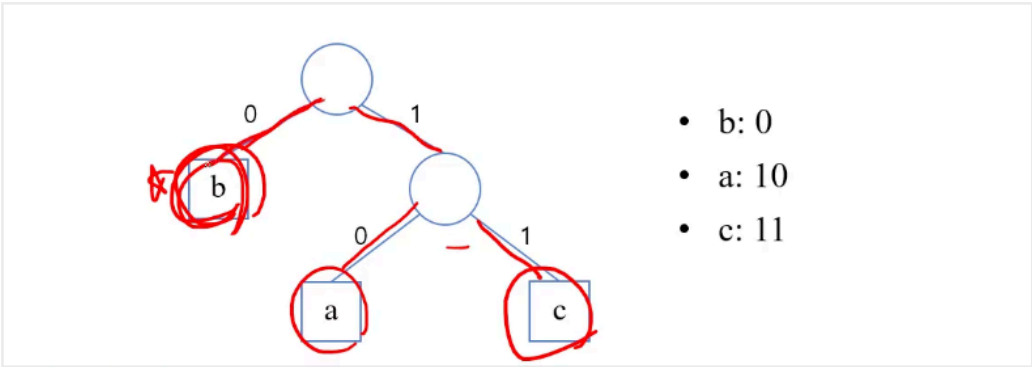
전치 코드

길이가 변하는 이진코드의 특수한 형태

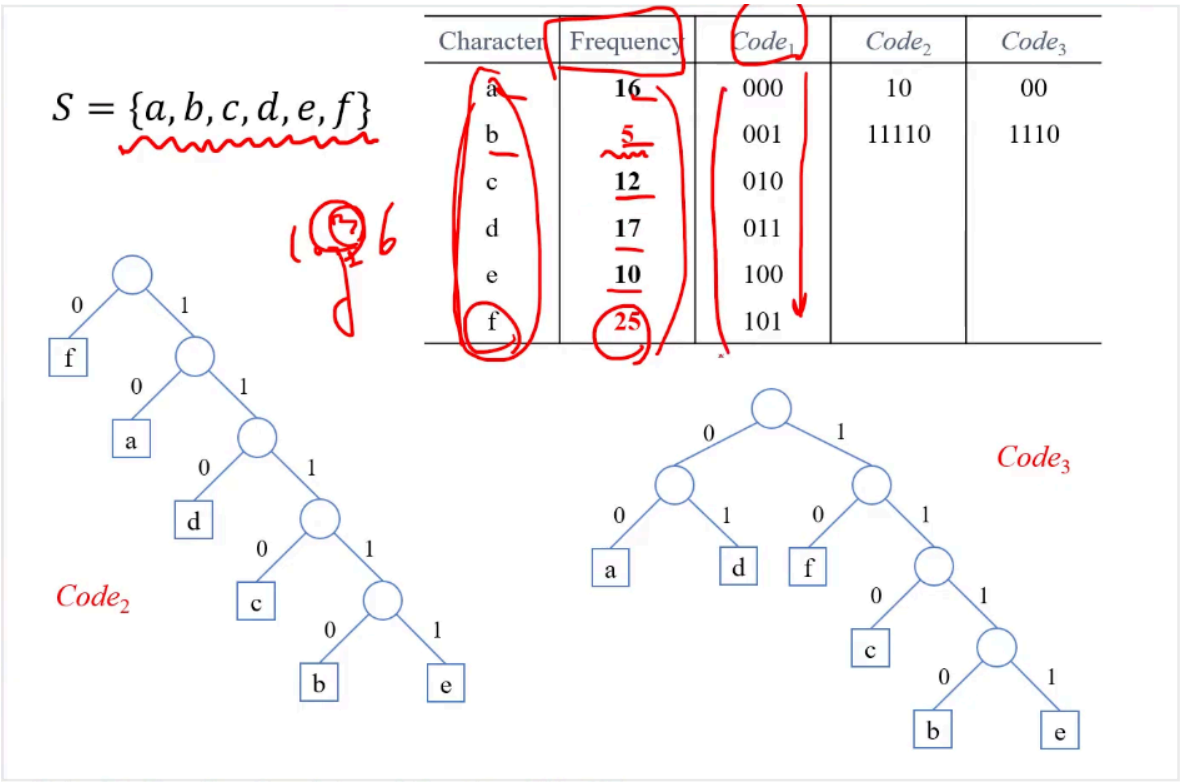
한 문자의 코드워드가 다른 문자의 코드워드의 앞부분이 될 수 없다

ex) 01 이 'a'의 코드워드라면 011은 'b'의 코드워드가 될 수 없다

전치코드의 표현 : 모든 전치코드는 리프노드가 코드 문자인 이진트리로 표현가능



장점 : 파일을 파싱(디코딩) 할 때 뒷부분을 볼 필요가 없다



허프만 코드

허프만 알고리즘에 의해 생성된 최적 이진코드

허프만 알고리즘

허프만 코드에 해당하는 최적 이진트리를 구축하는 그리디 알고리즘

최적 이진코드를 위한 이진트리의 구축

데이터 파일을 인코딩하는데 필요한 비트의 수 계산

주어진 이진트리를 T라고 하자

{v1,v2,...,vi) : 문자 vi 가 파일 내에서 나타나는 빈도수

depth(vi) : 이진트리 T에서 문자 vi노드의 깊이

$$bits(T) = \sum_{i=1}^{n} frequency(v_i)depth(v_i)$$

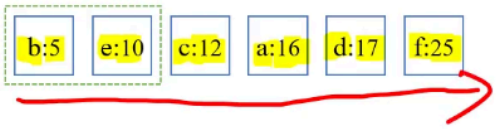
- $\text{bits}(T_2) = 231$
- $\text{bits}(T_3) = 212$

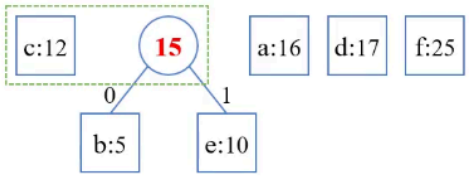
허프만 알고리즘: 탐욕 알고리즘

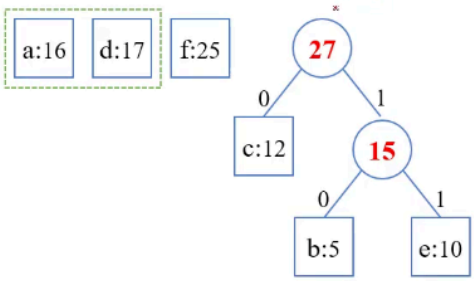
n: 주어진 데이터 파일내 문자의 개수

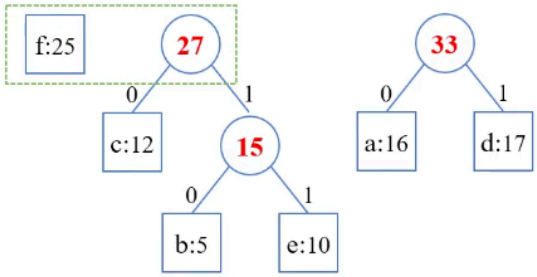
PQ: 빈도수가 낮은 노드를 먼저 리턴하는 우선순위큐(min-heap)

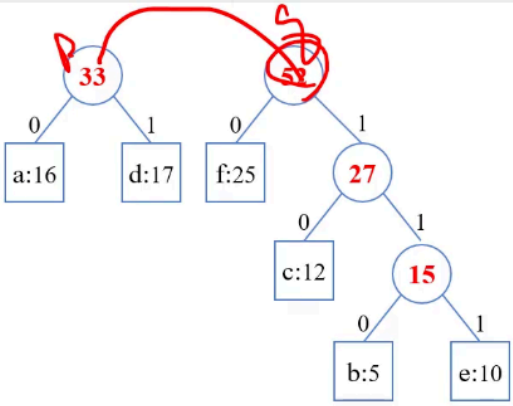
```
for i in [1..n-1]:
    remove(PQ, p)
    remove(PQ, q)
    r = new node
    r->left = p
    r->right = q
    r->frequency = q->frequency + q->frequency
    insert(PQ, r)
remove(PQ, r)
return r
```

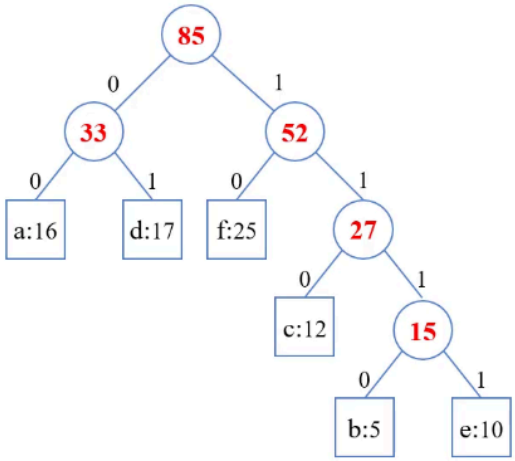

b:5
e:10
c:12
a:16
d:17
f:25


c:12
15
a:16
d:17
f:25
0
1
b:5
e:10


a:16
d:17
f:25
27
0
1
c:12
15
0
1
b:5
e:10


f:25
27
0
1
c:12
15
0
1
b:5
e:10
33
0
1
a:16
d:17


33
52
0
1
0
1
a:16
d:17
f:25
27
0
1
c:12
15
0
1
b:5
e:10


85
0
1
33
52
0
1
0
1
a:16
d:17
f:25
27
0
1
c:12
15
0
1
b:5
e:10