허프만 압축 알고리즘 및 기타 압축 알고리즘 성능비교

20120302 김우진
20130870 이건희

목차

- 허프만 압축 개요
- 허프만 압축 알고리즘
- 허프만 압축 결과
- 기타 압축 알고리즘과의 성능 비교

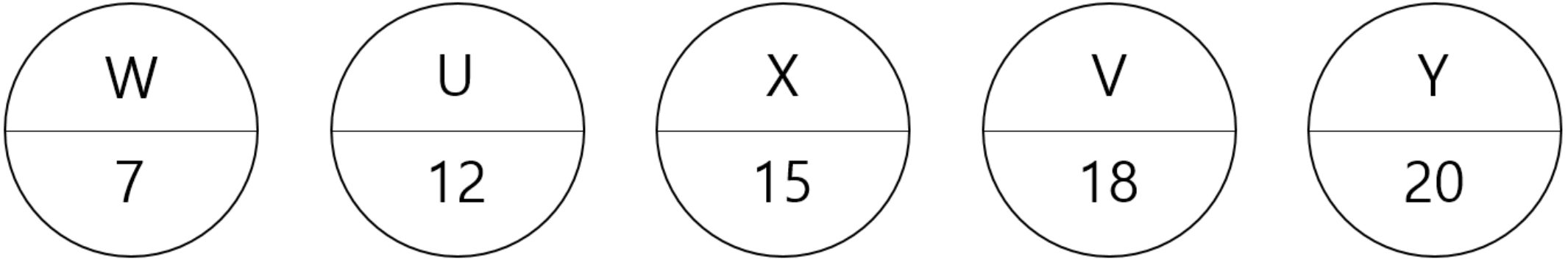
허프만 압축 개요

- 무손실 압축에 쓰이는 엔트로피 부호화의 일종
- 자료 압축의 가장 오래되고 기초적인 방법 중 하나

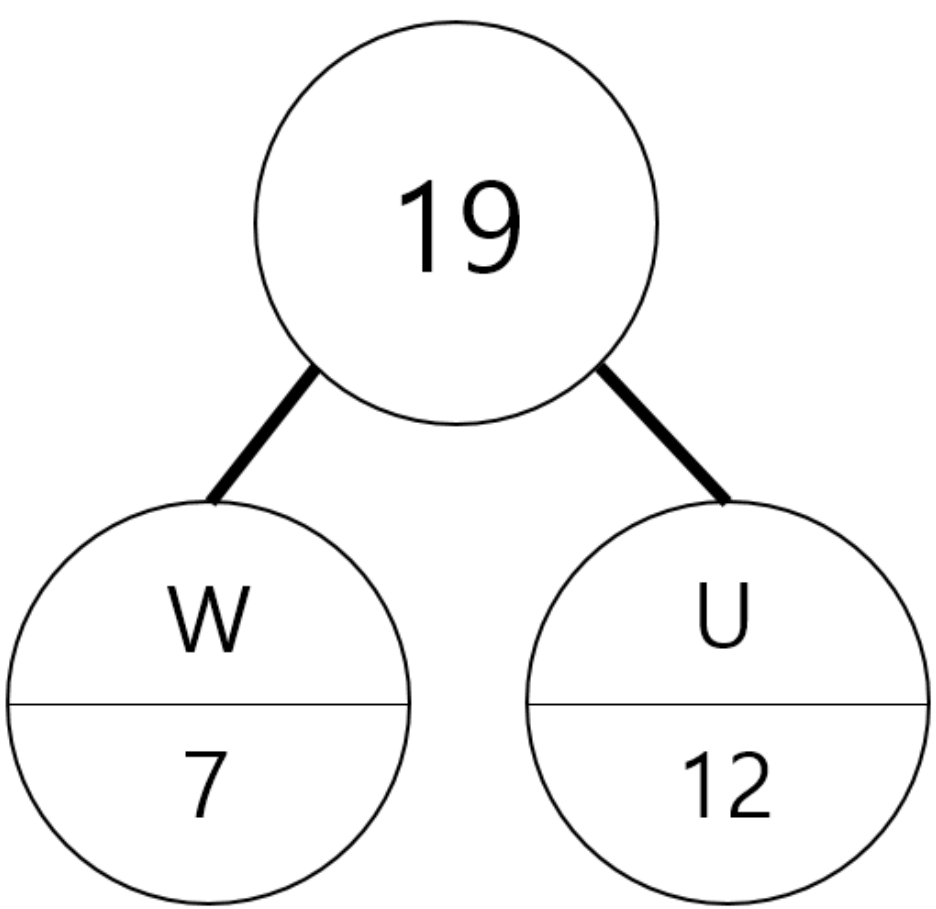
- 더 많이 쓰이는 문자일수록 비트의 수는 적어짐

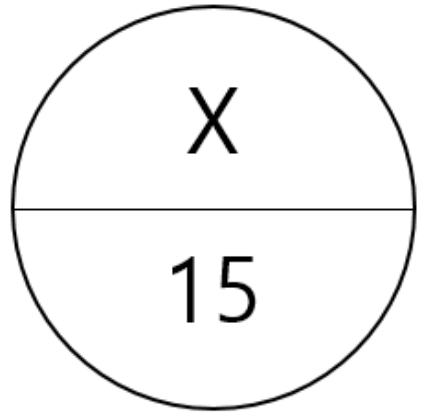
허프만 압축 알고리즘

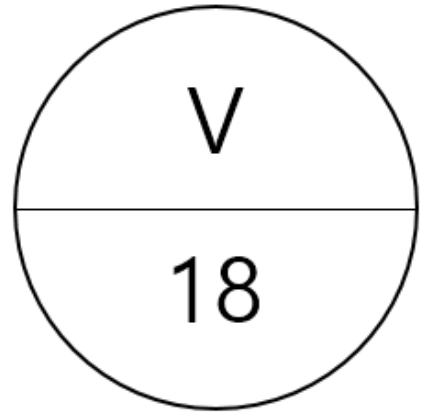
1. 문자가 적은 순서부터 정렬을 한다.

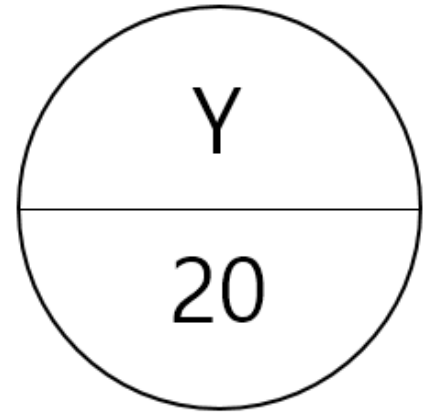


2. W와 U를 갖는 트리를 합친다.

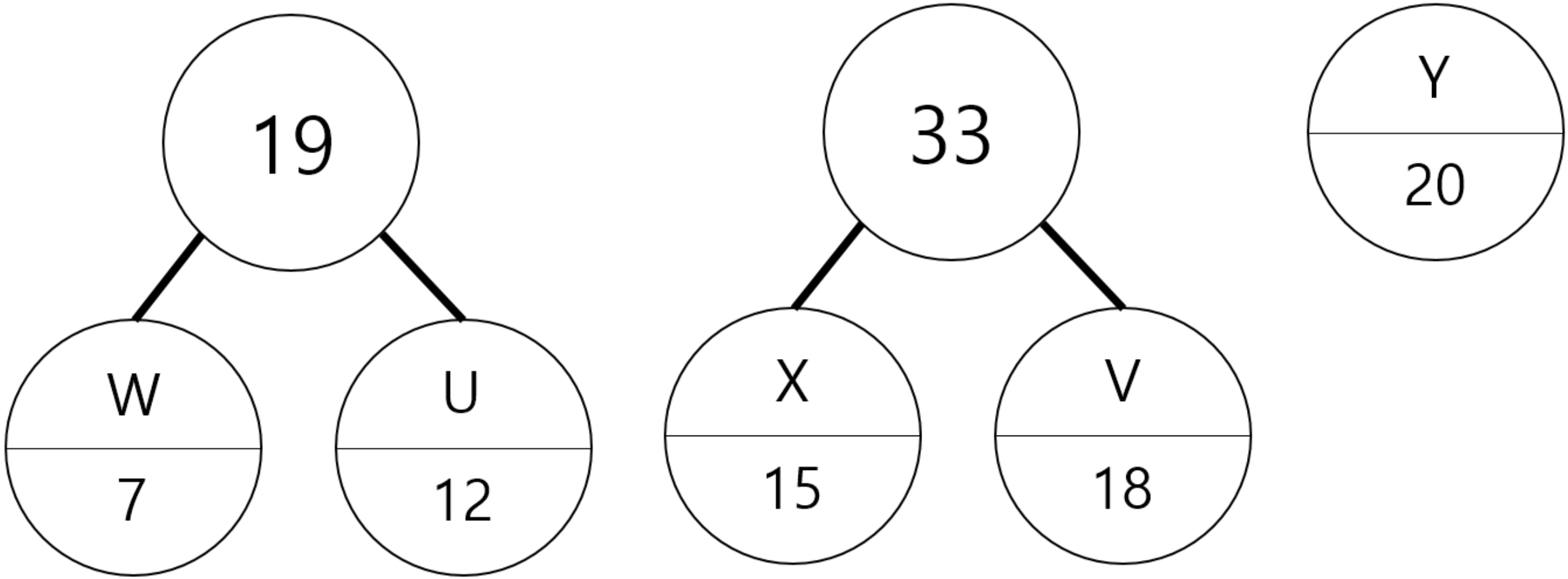




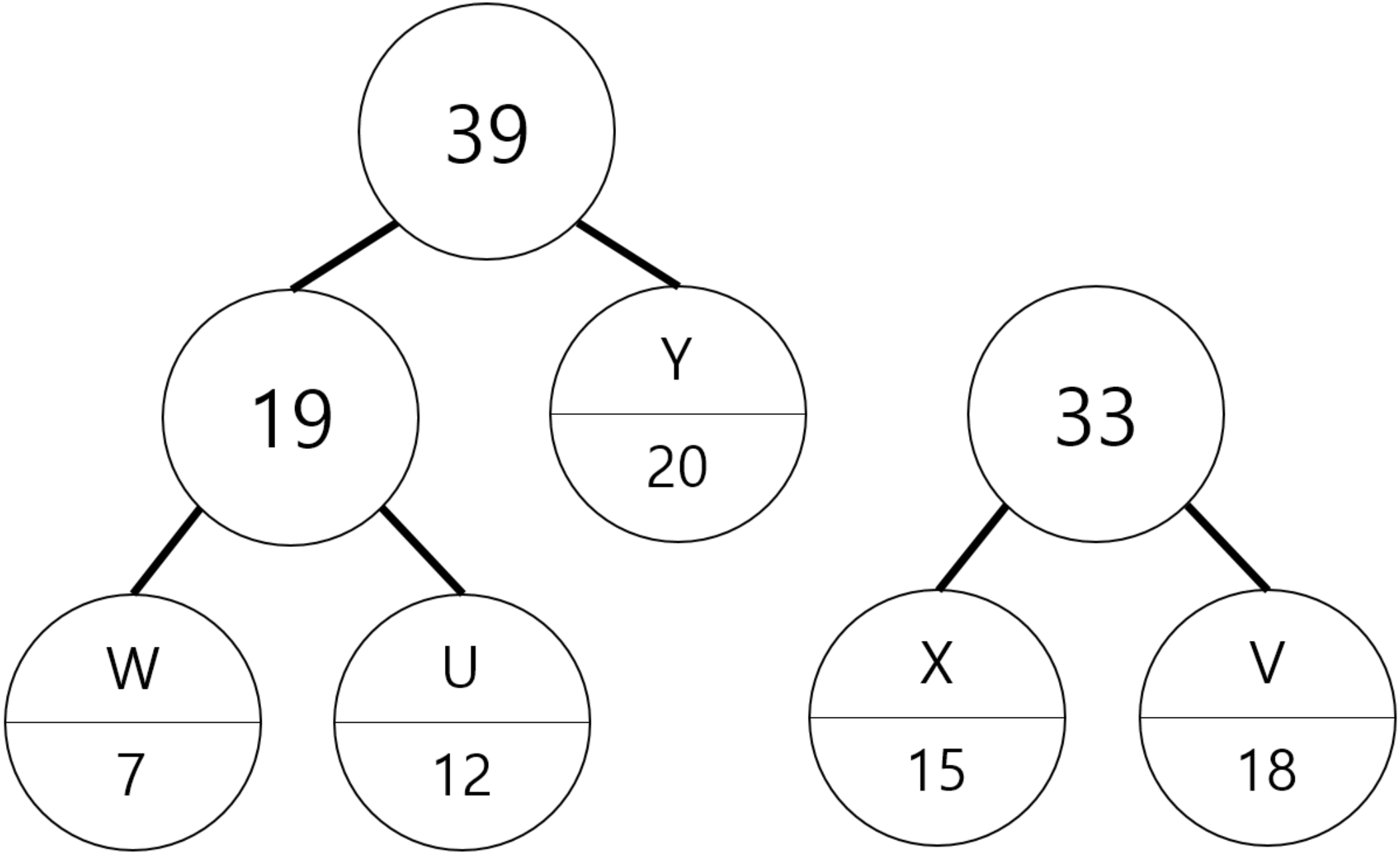




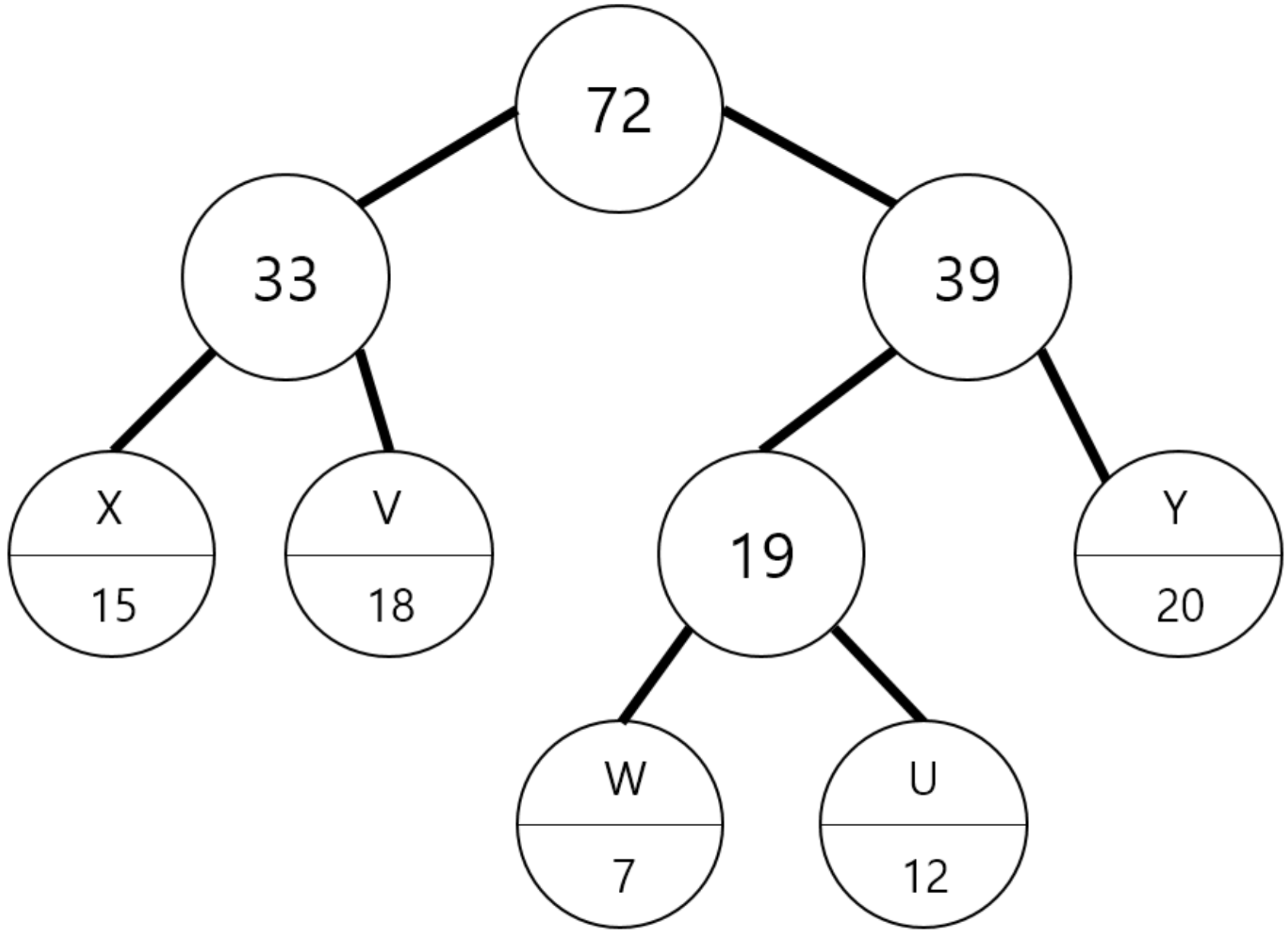
3. X와 V를 갖는 트리를 합친다.



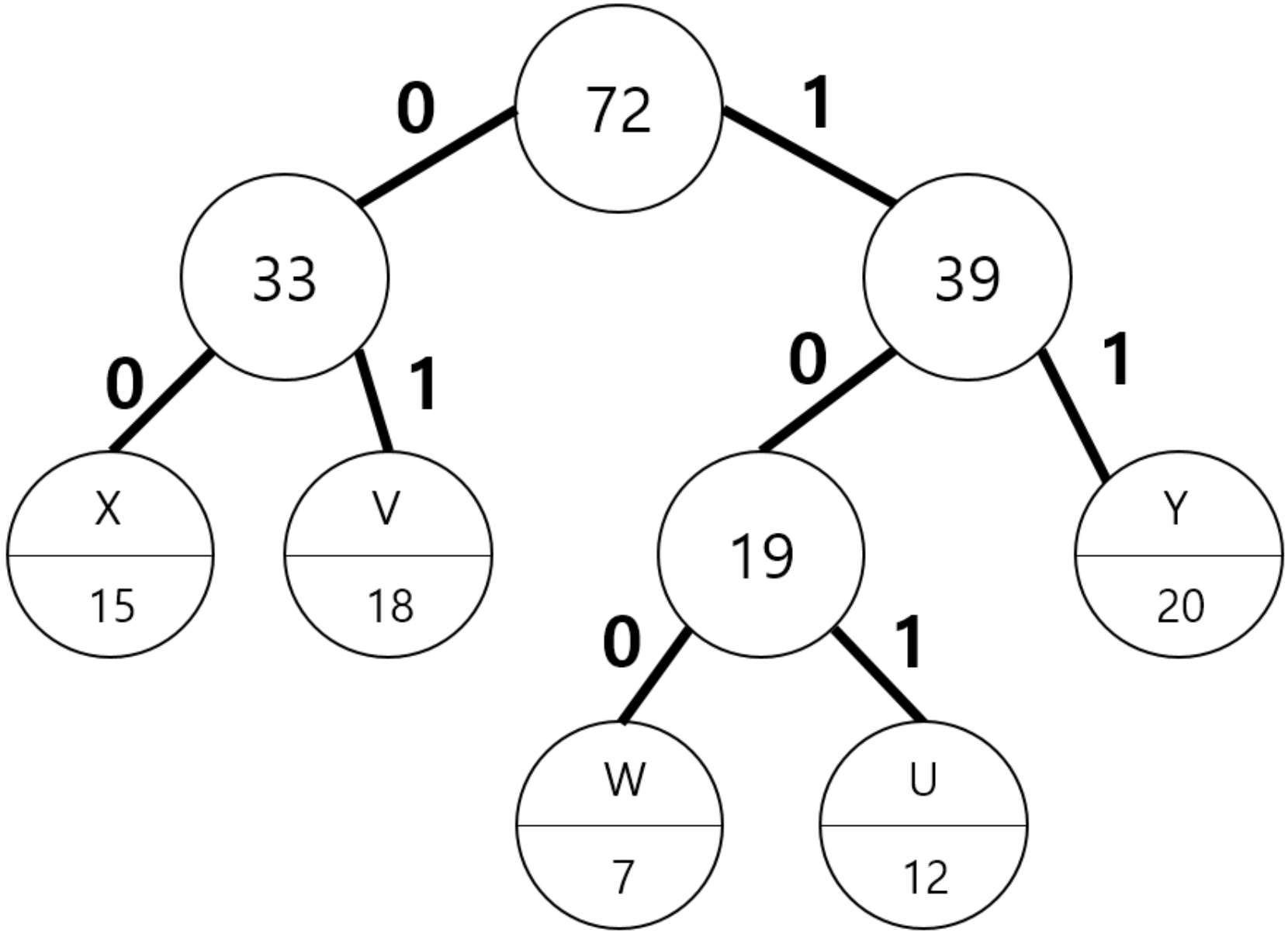
4. WU와 Y를 갖는 트리를 합친다.



5. WUY와 XV를 갖는 트리를 합친다.



6. 허프만 코드 생성



허프만 압축 결과

- **Data 1.** 일반적인 텍스트 – 압축효율 45.019%

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> news.txt

---------- Compress Result----------
Compressed file path: news_compress.bin
Compressed file size: 1012 bytes
Compressed ratio: 46.027 %

---------- Decompress Result----------
Decompressed file path: news_compress_restore.txt
Decompressed file size: 1875 bytes
Restoration ratio : 100.000 %
```

허프만 압축 결과

- **Data 1.** 일반적인 텍스트 – 압축효율 40.293%

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> test.txt

---------- Compress Result----------
Compressed file path: test_compress.bin
Compressed file size: 15534 bytes
Compressed ratio: 40.293 %

---------- Decompress Result----------
Decompressed file path: test_compress_restore.txt
Decompressed file size: 26017 bytes
Restoration ratio : 100.000 %
```

허프만 압축 결과

- **Data 2.** 특정 문자들로만 구성된 문서 – 압축효율 75.040%

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> ABC.txt

---------- Compress Result----------
Compressed file path: ABC_compress.bin
Compressed file size: 240623 bytes
Compressed ratio: 75.040 %

---------- Decompress Result----------
Decompressed file path: ABC_compress_restore.txt
Decompressed file size: 964014 bytes
Restoration ratio : 100.000 %
```

허프만 압축 결과

- **그 외의 테스트 케이스 적용**
  - 전반적으로 40~47% 정도의 효율을 보임.
  - 하지만 문자의 수가 적으면 적을 수록 압축 효율이 높아짐

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> news.txt

---------- Compress Result----------
Compressed file path: news_compress.bin
Compressed file size: 1012 bytes
Compressed ratio: 46.027 %

---------- Decompress Result----------
Decompressed file path: news_compress_restore.txt
Decompressed file size: 1875 bytes
Restoration ratio : 100.000 %
```

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> sample.txt

---------- Compress Result----------
Compressed file path: sample_compress.bin
Compressed file size: 180782 bytes
Compressed ratio: 45.019 %

---------- Decompress Result----------
Decompressed file path: sample_compress_restore.txt
Decompressed file size: 328809 bytes
Restoration ratio : 100.000 %
```

```
C:₩Python35₩python.exe C:/Python/Huffman_encoding/main.py
Input filename >> hello.txt

---------- Compress Result----------
Compressed file path: hello_compress.bin
Compressed file size: 46822 bytes
Compressed ratio: 65.110 %

---------- Decompress Result----------
Decompressed file path: hello_compress_restore.txt
Decompressed file size: 134198 bytes
Restoration ratio : 99.999 %
```

기타 압축 알고리즘과의 비교

	Huffman	LZH	7z	xz	tar	gz	zip
hello.txt	66.110%	99.996%	99.997%	99.998%	-1.486%	99.996%	99.994%
news.txt	46.027%	49.333%	99.989%	44.633%	-91.2%	60.133%	43.093%
sample.txt	46.019%	99.000%	99.694%	99.622%	-0.590%	99.002%	98.961%
test.txt	40.293%	91.632%	91.805%	92.143%	-6.268%	91.632%	91.126%
ABC.txt	76.040%	99.666%	99.967%	99.967%	-1.678%	99.678%	99.706%