# Word2Vec

# Task 1:

Write a function that takes list of words containing duplicates and
returns a list of words with no duplicates sorted by decreasing frequency in python

In [1]:

```
from collections import Counter

def sort_words_by_frequency(word_list):
    # count the frequency of each word in the list
    word_counts = Counter(word_list)

    # sort the unique words by decreasing frequency
    unique_words = word_counts.keys()
    sorted_words = sorted(unique_words, key=lambda x: word_counts[x], reverse=True)

    # create a list of tuples containing each unique word and its frequency count
    word_frequency = [(word, word_counts[word]) for word in sorted_words]

    return word_frequency
```

In [2]:

```
words = ["apple", "banana", "cherry", "mango" , "apple", "cherry", "cherry",
         "banana", "apple", "banana", "banana" , "apple" , "apple" ,"orange" , "mango"
word_frequency = sort_words_by_frequency(words)
for word, frequency in word_frequency:
    print(f"{word}: {frequency}")
```

```
apple: 5
banana: 4
cherry: 3
orange: 3
mango: 2
```

# Task 2:

Write a function that takes a text and a vocabulary as it's arguments
and returns set of words that appear in the text,
but not in the vocabulary. Both arguments can be represented as list of strings.

In [3]:

```
def find_unknown_words(text, vocabulary):
    # convert the text and vocabulary to sets for efficient comparison
    text_set = set(text)
    vocab_set = set(vocabulary)

    # find the words in the text that are not in the vocabulary
    unknown_words = text_set - vocab_set

    return unknown_words
```

In [4]:

```
text = "the quick brown fox jumps over the lazy dog"
vocabulary = ["the", "quick", "brown", "fox", "jumps"]
unknown_words = find_unknown_words(text.split(), vocabulary)
print(unknown_words)
```

```
{'lazy', 'over', 'dog'}
```