Exercise - 1:

```
import nltk
from nltk.wsd import lesk
from nltk.corpus.import .wordnet as wn.
nltk.download('wordnet')
for ss in wn.synsets('bass'):
    print(ss, ss.definition()).
print(lesk('I went fishing for some sea bass'.split(),
            'bass', 'n')).

print(lesk('Avishai Cohen is an Israeli jazz musician.
    He plays double bass and is also a composer'.split(),
    'bass', 'n')).
```

Exercise-2

```
for ss in wn.synsets('chair'):
    print(ss, ss.definition())
syn = wn.synsets('chair')[0].
print(syn).
print("synset name: ", syn.name()).
print("\n synset abstract term: ", syn.hypernyms())
print("\nsynset specific term:",
            syn.hypernyms()[0].hyponyms())
syn.root_hypernyms().
print("\nsynset root hyperterm:", syn.root_hypernyms)
```

Natural Language Processing Lab
Lab14. Word Sense Disambiguation with Improved Lesk Algorithm

In this lab, you will learn and disambiguate sentences with the correct senses using NLTK.

EXERCISE-1

Consider three examples of the distinct senses that exist for the word "bass":

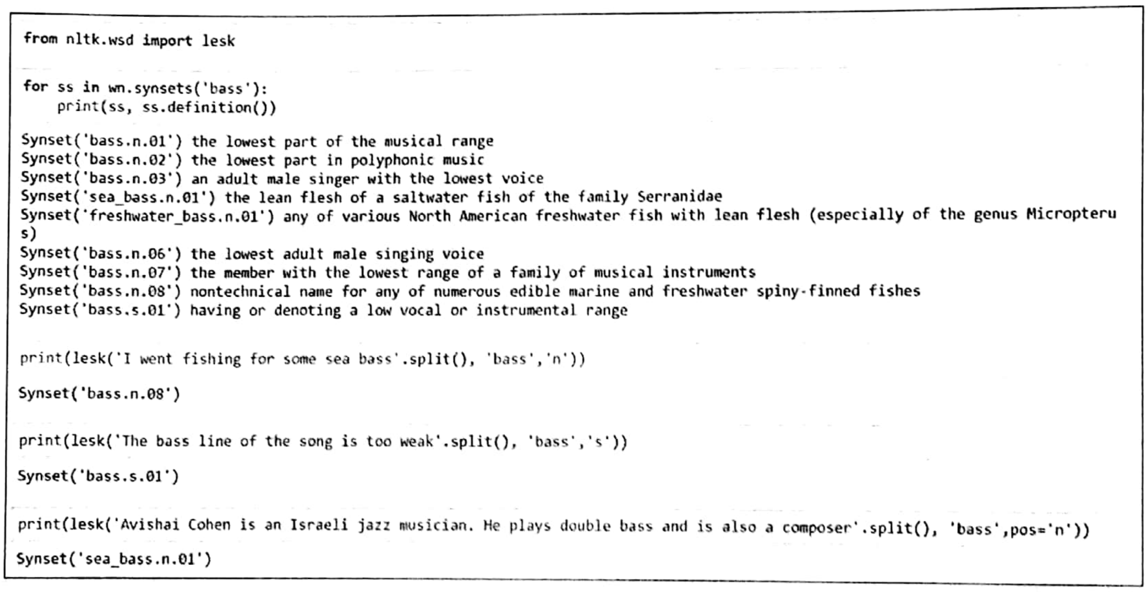
- a type of fish
- tones of low frequency
- a type of instrument

Consider the sentences:

- I went fishing for some sea bass.
- The bass line of the song is too weak.

Lesk algorithm syntax:

```
lesk(context_sentence, ambiguous_word, pos=None, synsets=None)
```



```
from nltk.wsd import lesk

for ss in wn.synsets('bass'):
    print(ss, ss.definition())

Synset('bass.n.01') the lowest part of the musical range
Synset('bass.n.02') the lowest part in polyphonic music
Synset('bass.n.03') an adult male singer with the lowest voice
Synset('sea_bass.n.01') the lean flesh of a saltwater fish of the family Serranidae
Synset('freshwater_bass.n.01') any of various North American freshwater fish with lean flesh (especially of the genus Micropteru
s)
Synset('bass.n.06') the lowest adult male singing voice
Synset('bass.n.07') the member with the lowest range of a family of musical instruments
Synset('bass.n.08') nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes
Synset('bass.s.01') having or denoting a low vocal or instrumental range

print(lesk('I went fishing for some sea bass'.split(), 'bass','n'))

Synset('bass.n.08')

print(lesk('The bass line of the song is too weak'.split(), 'bass','s'))

Synset('bass.s.01')

print(lesk('Avishai Cohen is an Israeli jazz musician. He plays double bass and is also a composer'.split(), 'bass',pos='n'))

Synset('sea_bass.n.01')
```

Note: For the third sentence where the bass is in the context of musical instrument, it is estimating the word as Synset('sea_bass.n.01) which is clearly not correct!

EXERCISE-2: Print senses for 'chair'

According to WordNet, how many distinct senses does 'chair' have? What are the hyponyms of 'chair' in its 'chair.n.01' sense? What is its hypernym, and what is its hyper-hypernym?

EXERCISE-3: *Disambiguate the correct senses given the context sentence*

```
from nltk.corpus import wordnet as wn
from nltk.stem import PorterStemmer
from itertools import chain

bank_sents = ['I went to the bank to deposit my money',
'The river bank was full of dead fishes']
```

Exercise : 3

```
from nltk.Corpus.Import wordnet as wn.
from nltk.stem Import PorterStemmer.
from Itertools Import chain.
bank_sents = ['I went to the bank to deposit my money',
        'the river bank was full of dead fishes'].
plant_sents = ['the workers at the Industrial plant were
        overworked', 'the plant was no longer bearing
        flowers'].
    ps = porterstemmer()
def. my_lesk(context_sentence, ambiguous_word, pos=None, stem=True,
    hyperhypo=True):
        max_overlaps = 0.
        lesk_sense=None.
    Context_sentence= Context_sentence.split()
    for ss in wn.synsets(ambigerous_word):
        # If pos Is specified.
        If pos and ss.pos Is not pos:
            Continue.
    lesk_dictionary = [ ].
    # Includes definition.
    defns = ss.definition().split()
    lesk_dictionary += defns.
    # Includes lemma_names
    lesk_dictionary += ss.lemma_names()
```

```
plant_sents = ['The workers at the industrial plant were overworked',
'The plant was no longer bearing flowers']

ps = PorterStemmer()

# define a function my_lesk
def my_lesk(context_sentence, ambiguous_word,
                pos=None, stem=True, hyperhypo=True):

    max_overlaps = 0
    lesk_sense = None
    context_sentence = context_sentence.split()

    for ss in wn.synsets(ambiguous_word):
        # If POS is specified.
        if pos and ss.pos is not pos:
            continue

        lesk_dictionary = []

        # Includes definition.
        defns = ss.definition().split()
        lesk_dictionary +=  defns

        # Includes lemma_names.
        lesk_dictionary += ss.lemma_names()

        # Optional: includes lemma_names of hypernyms and hyponyms.
        if hyperhypo == True:
            hhwords = ss.hypernyms() + ss.hyponyms()
            lesk_dictionary += list(chain(*[w.lemma_names() for w in hhwords] ))

        # Matching exact words causes sparsity,  so lets match stems.
        if stem == True:
            lesk_dictionary = [ps.stem(w) for w in lesk_dictionary]
            context_sentence = [ps.stem(w)  for w in context_sentence]

        overlaps = set(lesk_dictionary).intersection(context_sentence)

        if len(overlaps) > max_overlaps:
            lesk_sense = ss
            max_overlaps = len(overlaps)

    return lesk_sense

# evaluate senses
print("Context:", bank_sents[0])
answer = my_lesk(bank_sents[0],'bank')
print("Sense:", answer)
print("Definition:",answer.definition)

print("Context:", bank_sents[1])
answer = my_lesk(bank_sents[1],'bank')
print("Sense:", answer)
print("Definition:", answer.definition)

print("Context:", plant_sents[0])
answer = my_lesk(plant_sents[0],'plant')
print("Sense:", answer)
print("Definition:",answer.definition)
```

EXERCISE-4: Learn further examples for synsets at
https://www.programcreek.com/python/example/91604/nltk.corpus.wordnet.synsets

```
# Optional: Includes lemma-names of hypernyms and hyponyms.
    if hyperhypo = True:
        hhwords = ss.hypernyms() + ss.hyponyms()
    lesk-dictionary += list(chain(*[w.lemma-names() for w in
                              hhwords]))
 # Matching exact words:-
        if stem = = True:
            lest_dictionary = [ps.stem(w) for w in lesk-dictionary]
      Context_sentence = [ps. stem(w) for w in context_sentence].
       overlaps = set(lesk-dictionary). intersection(context_sentence)
      if len(overlaps) > max_overlaps:
            lesk-sense = ss
       max-overlaps = len(overlaps)
        return lesk-sense.
```

Evalate Senses:

```
        print('Context', bank_ sents[0])
    answer = my_ lesk(bank_ sents[0], bank)
     print ("Sense:", answer)
      print(" Definition:", answer. definition)

 Print ("Context:", plant_sents[0])
answer = my_lesk (plant_sents[0], 'Plant')
  print (" Sense:", answer)
  Print ("Definition:", answer. definition)
```