Text mining
Pracownia 1
Zajęcia 1 i 2

Zadanie 1. (7p) Jednym ze źródłem synonimów mogą być początkowe fragmenty artykułów encyklopedycznych, przykładowo ten początek

> **Śmigłowiec** lub **helikopter** (gr. heliks, D. helikos – skręcony; pteron – skrzydło) – statek powietrzny cięższy od powietrza (aerodyna), który ...

pozwala łatwo domyśleć się, że śmigłowiec i helikopter to synonimy. Napisz program, który wyszukuje synonimy, analizując początkowe fragmenty artykułów Wikipedii (podane na stronie wykładu). Interesują nas jedynie polskie słowa, czyli nie uznajemy za synonim np. angielskiego tłumaczenia danego terminu. Plik z początkami artykułów, stanowiący dane wejściowe do Twojego programu, ma następujący format:

```
### <tytuł artykułu w jednym wierszu>
<fragment artykułu, również w jednym wierszu>
<pusty-wiersz>
```

Twój program powinien analizować plik wejściowy i wypisywać (do pliku lub na standardowe wyjście) raport, w którym w każdym wierszu znajdują się synonimiczne względem siebie pojęcia. Poniższy przykładowy raport nie bazuje na Wikipedii, pokazuje jedynie format wyniku:

```
amoniak # mocznik
helikopter # śmigłowiec
bakłażan # oberżyna # gruszka miłości
```

Przygotuj listę 20-40 znalezionych synonimów (z których jesteś zadowolony), jak również dowolną liczbę ńiesynonimów", czyli par słów (fraz), które przez Twój program (w początkowym stadium) były błędnie zakwalifikowane jako synonimy.

Zadanie 2. (3p) W pliku `znaki_wikipedii.txt` znajdują się wszystkie znaki polskiej Wikipedii (każdy raz). Używając typu znaku pogrupuj je w zbiory o tym samym typie. Podziel dodatkowo klastry zawierające litery (tak żeby alfabet wietnamski, rosyjski i grecki były w osobnych klastrach), opierając się na założeniu, że znaki z tego samego alfabetu mają zbliżony kod.

Wymyśl sposób nazywania klastrów. Jeżeli znajdziesz nieliterowe klastry o dużej wielkości, również je podziel (w miejscach, w których mamy dużą różnicę kodów między znakami).

Wypisz raport, w którym każdy klaster znajduje się w osobnym wierszu, zaczynającym się od nazwy klastra, po której są kolejne znaki oddzielone spacjami.

Zadanie 3. (4p) Napisz swój własny tokenizator, który korzysta z typu znaku Unikodu i najpierw dzieli po spacjach, a następnie usuwa z obrzeży doklejone znaki interpukcyjne. Porównaj jego działanie na pliku `cytaty.txt` z funkcją `word_tokenize` z NLTK.

Wypisz w czytelnej postaci wszystkie miejsca, w których te tokenizatory dają inne wyniki. Zastanów się, jak mógłby wyglądać tokenizator lepszy od obu z tego zadania.

Zadanie 4. (2p), $\star$ Znajdź w Internecie jakiś tokenizator, który działa (również) dla języka polskiego. Porównaj jego działanie z tokenizatorem z poprzedniego zadania.