sprawko123456

July 6, 2021

1 Sprawozdanie nr 2

1.1 Jan Solarz, MiS

1.1.1 Wstp teoretyczny:

– ***Definicja algorytmów dynamicznych***: stosowana przewanie do rozwizywania zagadnie optymalizacyjnych. Jest alternatyw dla niektórych zagadnie rozwizywanych za pomoc algorytmów zachannych. Wynalazc techniki jest amerykaski matematyk Richard Bellman.

-Definicja odlegoci Levenhsteina: metryka w przestrzeni cigów znaków, zdefiniowana poprzez:

1.wstawienie nowego znaku do napisu
2.usunicie znaku z napisu
3.zamian znaku w napisie na inny znak
4.odlegoci pomidzy dwoma napisami jest najmniejsza liczba dziaa prostych
5.przeprowadzajcych jeden napis na drugi.

Miara ta znajduje zastosowanie w przetwarzaniu informacji, danych w postaci cigów symboli: w maszynowym rozpoznawaniu mowy, analizie DNA, rozpoznawaniu plagiatów, korekcie pisowni (np. wyszukiwanie w spisie telefonicznym bdnie podanego nazwiska).

Do czsto piszc na klawiaturze robimy literówki. Wykorzystujc ten algorytm moemy podpowiedzie uytkownikowi o co mu chodzio. Jest to szczególnie istotne w sklepie internetowym.

informacje uzyskane z Wikipedia.org

```
In [1]: a = input("Podaj wyraz pierwszy: ")
        b = input("Podaj wyraz drugi: ")

        """implementacja dwóch okien na dwa interesujce nas sowa"""

        """#input:
        #a- pierwszy wyraz
        #b- drugi wyraz"""

        """algorytm wyliczajcy odlego edycyjn (rónice dwóch wprowadzonych sów
        zgodnie z okrelonymi kryteriami punktowymi)"""

        #dodanie litery: rónica jednopunktowa
        #zmiana litery na inn: rónica dwupunktowa
```

```
        def odlego(a,b):
            n = len(a)
            m = len(b)
            if n > m:
                a,b = b,a
                n,m = m,n

            aktualny = range(n+1)

            for i in range(1,m+1):
                poprzedni = aktualny
                aktualny =  [i]+[0]*m
                for j in range(1,n+1):
                    dodaj = poprzedni[j]+1
                    usu = aktualny[j-1]+1
                    zmie = poprzedni[j-1]
                    if a[j-1] != b[i-1]:
                        zmie = zmie + 2
                    aktualny[j] = min(dodaj, usu, zmie)

            return aktualny[n]

        print(f"Odlego edycyjna wynosi: {odlego(a,b)}")

        """#output:
        liczba, która jest dan odlegoci""";
```

```
Podaj wyraz pierwszy: fasdvasd
Podaj wyraz drugi: aavsdcsg
Odlego edycyjna wynosi: 8
```

```
In [3]: #dowolno okrelenia parametru podobiestwa
        o = input("Podaj podobiestwo: ")
        if ValueError:
            print('Warto musi by wpisana!');
        elif float(o)>1 or float(o)<0:
            print('Zy parametr podobiestwa');


        """#input:
        #sowo- badany przez nas wyraz (wyjciowe)
        #sownik- sownik wyrazów,które bd przez nas porównywane do sowa wyjciowego"""
        def znajd(sowo, sownik):
            n = len(sowo)
            sowo = sowo.lower()
            wyniki = [(odlego(sowo, s.lower())/max(n, len(s)), s) for s in sownik]
            wyniki = [(d, s) for d, s in wyniki if d<=float(o)]
```

```
            wyniki.sort()
            wyniki = [s for d, s in wyniki]
            return wyniki
        """#output:
        #sowo wyjciowe
        #wyrazy które speniaja kryteria podobiestwa""";
```

```
Podaj podobiestwo: 0.7
Warto musi by wpisana!
```

```
In [22]: znajd("gitara", ["gitarzysta", "gitarowy", "git", "gatunek", "stara"])
```

```
Out[22]: ['gitarzysta', 'git', 'gitarowy', 'stara']
```

```
In [23]: znajd("komputer", ["komp", "komputerownia", "kompost", "komputerowy"])
```

```
Out[23]: ['komputerowy', 'komputerownia', 'komp']
```

```
In [24]: znajd("piwo", ["piwko", "piwerko", "piteczek", "piwuncio", "programowanko"])
```

```
Out[24]: ['piwko', 'piwerko', 'piwuncio']
```

```
In [25]: znajd("nauka", ["naukowy", "naukowiec", "kaukaz"])
```

```
Out[25]: ['kaukaz']
```

```
In [28]:  znajd("abcdef", ["iuahjbcdef", "khagbcd", "abcd", "pakbckdlef"])
```

```
Out[28]: ['abcd', 'iuahjbcdef', 'pakbckdlef']
```

1.1.2 Wnioski:

- Sowa bardzo czsto mog wyglda na pozór zupenie inaczej
- dziki temu algorytmowi mona zrozumie na jakiej zasadzie dziaa powszechnie uywana przegldarka Google

```
In [ ]:
```