Algorytmy i Funkcje Python

Algorym Euklidesa

Zacznijmy sobie od podstawowego algorytmu na najwiekszy wspólny dzielnik

[ ]:
```python
def NWD(a,b):
    while a != b:
        if a > b:
            a -= b
        else:
            b -= a
    return a # tutaj jest obojętne co zwracamy ponieważ a == b
```

należy również pamiętać o zoptymalizowanej wersji tego algorymu, czas pojawiał się w pytaniach maturalnych

[ ]:
```python
def Zotpymalizowane_NWD(a,b):
    while b:
        temp = a
        a = b
        b = temp % b
    return a # tutaj ważne jest aby wzrócić a ponieważ b będzie równe 0
```

liczba pierwsza

następnym algorymem będzie sprawdzenie czy liczba jest pierwsza

[ ]:
```python
def czy_pierwsza(liczba):
    if liczba == 2:
        return True

    for i in range(2,liczba):
        if liczba % i == 0:
            return False # w niektórych algorytmach lepiej jest sprawdzać czy nasze pytanie jest fałszywe
    return True
```

jest to najdłuższa wersja algorytmu, żeby go zoptymalizować należy sprawdzać dzieliki do liczby/2 a najlepiej do pierwiastka liczby

jeśli jednak ten kod jest wciąż zbyt wolny należy zastosować sito erastotenesa

[ ]:
```python
przedział = 100000 # listę tworzymy od 0 do wybranej przez nas liczby
lista = [] # w tej liście 1 będzie oznaczać liczbę nieskreśloną, a 0 skreśloną
for i in range(przedział):
    lista.append(1)

lista[0]= 0
lista[1] = 0# ponieważ 1 i 0 nie są liczbami pierszymi

#wykreślanie
for i in range(len(lista)):
    if lista[i] == 1:
        for p in range(i*2, len(lista), i):
            lista[p] = 0

print(lista) # w tej liście index oznacza liczbę a 0/1 czy jest liczbą pierwszą czy nie
```

liczba dziesiętna na inny system liczbowy

kiedy będziemy potrzebowalić liczbę dziesiętną na inny system liczbowy niż funkcje na to pozwalają przyda się nam owa funkcja:

[ ]:
```python
def zamiana_systemu(sys,liczba):
    cyfry = '0123456789ABCDEF' #zebysmy wiedzieli co wpisac jesli system jest wiekszy niz 10
    liczba_w_systremie = '' # liczba zamieniony przez nasz system

    while liczba > 0: # wykonujemy to aż nasza liczba nie będzie zerem
        cyfra = liczba % sys # to co było po prawej stronie kreski jak pisaliśmy to na kartce
        liczba_w_systremie += cyfry[cyfra] #index od razu mowi nam jaki to jest znak jesli system jest wiekszy niz 10
        liczba = liczba // sys # lewa strona od kreski

    liczba_w_systremie = liczba_w_systremie[::-1] #odwrocenie pod koniec
    return liczba_w_systremie
```

pierwiastek

możemy czasem potrzebować pierwiastka funkcji, lub jego przybliżenia całkowitego

[ ]:
```python
def pierwiastek(liczba):
    i = 1
    while i*i < liczba:
        i += 1
    return i #należy pamiętać o tym że i może być wartością przybliżoną w góre jeśli liczba nie ma pierwiastka
```

liczenie elementów

jeśli chcemy policzyć elementy w liście, bądz litery w tekscie przydadzą się nam słowniki

[26]:
```
lista = ['a','b','a','t','b','slowo',4,7,'a',4]
slownik = {}
for el in lista:
    slownik[el] = 0 #najpierw musimy zadeklarować wartości

for el in lista:
    slownik[el] += 1 #zwiększmy o 1 za każde wystąpienie

print(slownik)
```

```
{'a': 3, 'b': 2, 't': 1, 'slowo': 1, 4: 2, 7: 1}
```

przydatne funkcje

ord() i chr()

ord(litera) = kod ascii litery

chr(ascii) = litera

[14]:
```
print(ord('a'))
print(ord('A')) # trzeba pamiętać że mała litera ma inny kod ascii niż duża
print(20*'-')
print(chr(97))
print(chr(65))
```

```
97
65
--------------------
a
A
```

abs()

abs przydaje się w momędcie kiedy potrzebujemy wartości bezwzględnej

[15]:
```
wynik = abs(30-100)
liczba = abs(-50)
print(wynik)
print(liczba)
```

```
70
50
```

set()

set() jest używane do przechowywania niepowtarzalnych zmiennych

[16]:
```
thisset = {"apple", "banana", "cherry" , "apple"}
print(thisset)
```

```
{'cherry', 'banana', 'apple'}
```

listy równierz możemy przekształcić w seta i wyeliminować powtórzenia

[19]:
```
lista = [3,5,7,2,5,7,3,6,8,2,1,6,8]
set_listy = set(lista)
print(set_listy)
#możemy także zamienić seta na listę i mieć listę bez duplikatów
lista_bez_d = list(set_listy)
print(lista_bez_d)
```

```
{1, 2, 3, 5, 6, 7, 8}
[1, 2, 3, 5, 6, 7, 8]
```

przypisania wielu zmiennych

czasem zamiast kombinowania z pomianą albo przypisywaniem wartości możemy zmienna zapisać w następujący sposób:

[20]:
```
a,b = 3,2
print(a,b)
#zamiana
a, b = b, a
print(a,b)
```

```
3 2
2 3
```

inny system liczbowy na dziesiętny

jeśli potrzebujemy zamienić liczbę z innego systemu liczbowego na dziesiętną robimy to następująco:

[23]:
```
liczba_czworkowa = '2113' # liczba w innym systemie musi być stringiem
l_dziesietna = int(liczba_czworkowa,4) # 4 oznacza w jakim systemie była liczba
print(l_dziesietna)

liczba_bin = '1010101'
l_dz = int(liczba_bin, 2)
print(l_dz)
```

```
151
85
```

count() oraz index()

count służy nam do policzenia ile dany element razy znajduje się w liście

index na tomiast znajduje nam najbliży index danego elementu

[30]:
```
lista = [3,5,7,2,5,7,3,6,7,8,2,1,6,8]
liczba_7 = lista.count(7)
print(liczba_7)

najblizsza_8 = lista.index(8)
print(najblizsza_8)
```

```
3
9
```

odwracanie listy

slicowanie list pozwala nam na odwrócenie listy jeśli ustawimy krok ujemnty

[31]:
```
slowo = 'matura'
odwrocenie = slowo[::-1]
```

przydaje się kiedy chcemy sprawdzić czy słowo jest palindromem

startswith() i endswith()

są to zapytania czy dany ciąg kończy liste/słowo

[42]:
```
slowo = 'ania ma kota'
print(slowo.endswith('ota'))

print(slowo.startswith("ma"))
```

```
True
False
```