Danmarks Tekniske Universitet

Kursusnavn	Introduktion til programmering og databehandling
Kursusnummer	02633
Hjælpemidler	Alle hjælpemidler er tilladt
Varighed	2 timer
Vægtning	Opgaverne vægtes ens

Indhold

Assignment A: Heron's formula 2
Assignment B: Area of a polygon 3
Assignment C: Fitting a polynomial 4
Assignment D: Predictability of a number sequence 5
Assignment E: Syllable counter 6

Aflevering

Du skal aflevere dine løsninger elektronisk:

1. Du kan uploade dine løsninger individuelt på CodeJudge (`dtu.codejudge.net/prog-jan17/assignment`) under *Afleveringer/Exam*. Når du afleverer en løsning på CodeJudge bliver det test-eksempel som fremgår i opgavebeskrivelsen kørt på din løsning. Hvis din løsning består denne ene test, vil den fremgå som *Submitted*. Det betyder at din løsning består på dette ene test-eksempel. Du kan uploade til CodeJudge så mange gange som du ønsker under eksamen.

2. Du skal uploade alle dine løsninger på CampusNet. Hver assignment skal uploades som en separat .py fil med samme navn som funktionen i opgaven:

   (a) `trianglearea.py`
   (b) `polygonarea.py`
   (c) `fitpolynomial.py`
   (d) `predictability.py`
   (e) `syllables.py`

   Filerne skal afleveres separat (*ikke* som en zip-fil) og skal have præcis disse filnavne.

Efter eksamen vil dine løsninger blive automatisk evalueret på CodeJudge på en række forskellige tests for at undersøge om de generelt fungerer korrekt. Bedømmelsen af din aflevering foretages udelukkende på baggrund af hvor mange af de automatiserede test der bestås.

- Du skal sikre dig at din kode følger specifikationerne nøje.
- Hver løsning må ikke indeholde nogen yderligere kode udover den specificerede funktion.
- Husk at du kan tjekke om dine løsninger følger specifikationerne ved at uploade dem til CodeJudge.
- Bemærk at alle vektorer og matricer anvendt som input eller output skal være numpy arrays.

Assignment A Heron's formula

Givet de tre sidelængder (a, b, og c) i en trekant, kan arealet A beregnes med Heron's formel,

$$A = \sqrt{p(p-a)(p-b)(p-c)},$$

hvor

$$p = \frac{1}{2}(a+b+c).$$

Formlen er kun gyldig når sidelængderne danner en mulig trekant, dvs. når følgende tre uligheder er opfyldt:

$$a+b>c, \quad a+c>b, \quad b+c>a.$$

Problemdefinition

Skriv en funktion med navnet `trianglearea` der tager de tre sidelængder i en trekant som input og beregner arealet ved hjælp af Heron's formel. Hvis de givne sidelængder ikke danner en mulig trekant skal funktionen returnere nul.

Løsningsskabelon

```python
def trianglearea(a, b, c):
  #insert your code
  return A
```

Input

`a`, `b`, `c`	Sidelængder (positive decimaltal).

Output

`A`	Areal af trekanten.

Eksempel

Forestil følgende input, $a = 4.5$, $b = 6$, og $c = 7.5$. Disse sidelænger opfylder alle tre uligheder, og udgør dermed en mulig trekant. Arealet beregnes da som

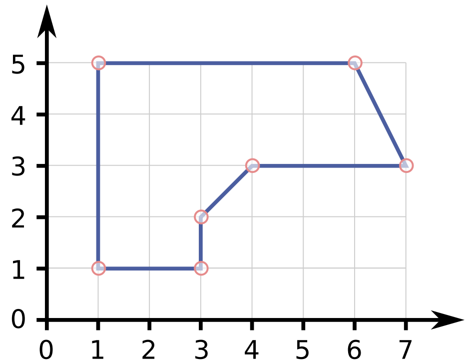
$$p = \frac{1}{2}(4.5+6+7.5) = 9$$

$$A = \sqrt{9(9-4.5)(9-6)(9-7.5)} = 13.5$$

A

Assignment B Area of a polygon

En polygon kan specificeres ved en liste af hjørne-koordinater $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. For eksempel kan polygonen i figuren nedenfor specificeres ved koordinaterne $\{(1,1), (3,1), (3,2), (4,3), (7,3), (6,5), (1,5)\}$.



På baggrund af koordinaterne kan arealet af polygonen beregnes vha. følgende formel

$$A = \frac{1}{2}\left(\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_4 \\ y_3 & y_4 \end{vmatrix} + \cdots + \begin{vmatrix} x_n & x_1 \\ y_n & y_1 \end{vmatrix} \right),$$

hvor $\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = (x_1 y_2 - y_1 x_2)$ betegner determinanten af den givne 2×2 matrix. Bemærk at formlen for arealet indeholder et determinant-led for hver kant i polygonen.

Bemærkning: Formlen er gyldig når polygonen ikke krydser sig selv og rækkefølgen af koordinaterne er givet mod urets retning (venstre om). I denne opgave antager vi at disse betingelser er opfyldt.

Problemdefinition

Skriv en funktion med navnet `polygonarea` der tager som input to vektorer med henholdsvis x- og y-koordinater, og returnerer arealet af polygonen.

Løsningsskabelon

```
def polygonarea(x, y):
  #insert your code
  return A
```

Input	
`x, y`	X- og y-koordinater der specificerer en polygon (vektorer med decimaltal).
Output	
`A`	Areal af polygonen.

Eksempel

Forestil dig følgende input-koordinater som svarer til polygonen i figuren.

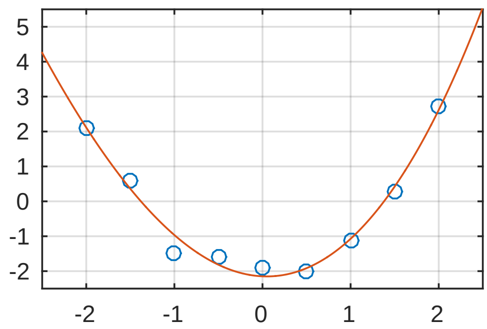
$$x = [1,\ 3,\ 3,\ 4,\ 7,\ 6,\ 1], \quad y = [1,\ 1,\ 2,\ 3,\ 3,\ 5,\ 5]$$

Arealet kan da beregnes som

$$\begin{aligned} A &= \frac{1}{2}\left(\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_4 \\ y_3 & y_4 \end{vmatrix} + \begin{vmatrix} x_4 & x_5 \\ y_4 & y_5 \end{vmatrix} + \begin{vmatrix} x_5 & x_6 \\ y_5 & y_6 \end{vmatrix} + \begin{vmatrix} x_6 & x_7 \\ y_6 & y_7 \end{vmatrix} + \begin{vmatrix} x_7 & x_1 \\ y_7 & y_1 \end{vmatrix} \right) \\ &= \frac{1}{2}\left(\begin{vmatrix} 1 & 3 \\ 1 & 1 \end{vmatrix} + \begin{vmatrix} 3 & 3 \\ 1 & 2 \end{vmatrix} + \begin{vmatrix} 3 & 4 \\ 2 & 3 \end{vmatrix} + \begin{vmatrix} 4 & 7 \\ 3 & 3 \end{vmatrix} + \begin{vmatrix} 7 & 6 \\ 3 & 5 \end{vmatrix} + \begin{vmatrix} 6 & 1 \\ 5 & 5 \end{vmatrix} + \begin{vmatrix} 1 & 1 \\ 5 & 1 \end{vmatrix} \right) \\ &= \frac{1}{2}\Big((-2) + 3 + 1 + (-9) + 17 + 25 + (-4) \Big) = 15.5 \end{aligned}$$

Assignment C Fitting a polynomial

Du får givet et sæt x- og y-punkter $\{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$ og du ønsker at finde et n'te-gradspolynomium som fitter data så godt som muligt. Du skal bestemme koefficienterne $a_0, \ldots, a_n$ i polynomiet $f_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ ved hjælp af mindste kvadraters metode: Først konstrueres $k \times (n+1)$ matricen W,



$$W = \begin{bmatrix} 1 & x_1^1 & \ldots & x_1^n \\ 1 & x_2^1 & \ldots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_k^1 & \ldots & x_k^n \end{bmatrix}.$$

Derefter findes koefficienterne ved brug af følgende formel, hvor vektoren $y = [y_1,\ y_2, \ldots, y_k]^\top$,

$$a = (W^\top W)^{-1}W^\top y.$$

Resultatet er vektoren a som har $n + 1$ elementer.

Problemdefinition

Skriv en funktion med navnet `fitpolynomial` der tager som input to vektorer med længde k indeholdende x- og y-punkterne, samt graden n af polynomiet. Funktionen skal returnere koefficient-vektoren a.

Løsningsskabelon

```python
def fitpolynomial(x, y, n):
  #insert your code
  return a
```

Input

`x, y`	Input- og output-punkter (vektorer med decimaltal, længde k).
`n`	Graden af polynomiet (positivt heltal).

Output

`a`	Polynomiets koefficienter (vektor med decimaltal).

Eksempel

Forestil dig følgende input (også illustreret i figuren), hvilket giver den følgende matrix W.:

$$x = [-2,\ -1.5,\ -1,\ -0.5,\ 0,\ 0.5,\ 1,\ 1.5,\ 2]^\top,$$
$$y = [2.1,\ 0.6,\ -1.5,\ -1.6,\ -1.9,\ -2,\ -1.1,\ 0.3,\ 2.7]^\top,$$
$$n = 3.$$

$$W = \begin{bmatrix} 1 & -2 & 4 & -8 \\ 1 & -1.5 & 2.25 & -3.375 \\ 1 & -1 & 1 & -1 \\ 1 & -0.5 & 0.25 & -0.125 \\ 1 & 0 & 0 & 0 \\ 1 & 0.5 & 0.25 & 0.125 \\ 1 & 1 & 1 & 1 \\ 1 & 1.5 & 2.25 & 3.375 \\ 1 & 2 & 4 & 8 \end{bmatrix}$$

Koefficienterne udregnet vha. formlen giver resultatet $a \approx [-2.1476,\ -0.1135,\ 1.1286,\ 0.0599]^\top$.

Assignment D — Predictability of a number sequence

En række af tal kan være mere eller mindre forudsigelig, forstået på den måde at det kan være let eller svært at gætte det næste tal i rækken givet de forgående tal. Her ser vi på en talrække med længde N som kun indeholder heltal mellem 1 og 5 (begge inkluderet), og vi definerer graden af forudsigelighed ifølge den følgende algoritme.

- Du er givet en sekvens af tal, $[x_1,\ x_2,\dots,\ x_N]$.
- Beregn en vektor q (med 5 elementer) hvori element $q[i]$ er antallet af gange heltallet i optræder i sekvensen.
- Beregn en matrix R (med dimension 5×5) hvori element $R[i,j]$ er antallet af gange heltallet j følger umiddelbart efter i i sekvensen.
- Beregn forudsigeligheden P ved følgende formel

$$P = \frac{1}{N}\left(\frac{q[x_1]}{N} + \sum_{n=2}^{N}\frac{R[x_{n-1},x_n]}{q[x_{n-1}]}\right).$$

Problemdefinition

Skriv en funktion med navnet `predictability` der tager en sekvens x af tal som input og returnerer forudsigeligheden P beregnet som beskrevet i algoritmen ovenfor.

Løsningsskabelon

```
def predictability(x):
  #insert your code
  return P
```

Input	
x	Sekvens af tal (vektor med heltal mellem 1 og 5).
Output	
P	Forudsigelighed (decimaltal).

Eksempel

Forestil dig følgende tal-sekvens med længde $N = 15$

$$x = [1,\ 5,\ 5,\ \uwave{3,\ 5},\ 1,\ \uwave{3,\ 5},\ \underline{\underline{4,}}\ 5,\ \underline{\underline{4,}}\ 1,\ 5,\ 5,\ \underline{\underline{4}}].$$

Vektoren q skal da være $q = [3,\ 0,\ 2,\ \underline{\underline{3,}}\ 7]$. For eksempel (vist med dobbelt understregning) har vi $q[4] = 3$ fordi tallet 4 optræder 3 gange. Matricen R kan beregnes til

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \uwave{2} \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 3 & 2 \end{bmatrix}$$

For eksempel (vist med bølget understregning) har vi $R[3,5] = 2$ fordi 5 følger umiddelbart efter 3 to gange i sekvensen. Endelig kan P beregnes som

$$P = \frac{1}{15}\left(\frac{3}{15}+\frac{2}{3}+\frac{2}{7}+\frac{1}{7}+\frac{2}{2}+\frac{1}{7}+\frac{1}{3}+\frac{2}{2}+\frac{3}{7}+\frac{1}{3}+\frac{3}{7}+\frac{1}{3}+\frac{2}{3}+\frac{2}{7}+\frac{3}{7}\right) \approx 0.4451$$

Assignment E Syllable counter

Hurtig-læsnings-software viser en tekst til en læser, ét ord ad gangen i høj hastighed. Ifølge en teori skal varigheden af visningen af et ord være proportionelt med antallet af stavelser N i ordet. I denne opgave skal du lave en funktion der (tilnærmet) estimerer antallet af stavelser i et ord, ifølge denne regel:

Antallet af stavelser er lig med antallet af vokaler i ordet, bortset fra at to eller flere vokaler i træk kun tælles som én, og en eller flere vokaler i slutningen af ordet ikke tælles med.

Du kan antage at input kun indeholder små bogstaver `a`–`z`, hvoraf `a`, `e`, `i`, `o`, `u`, og `y` er vokaler.

Problemdefinition

Skriv en funktion med navnet `syllables` der tager som input et streng indeholdende et enkelt ord, og returnerer antallet af stavelser estimeret efter ovenstående regel.

Løsningsskabelon

```
def syllables(word):
  #insert your code
  return N
```

Input	
`word`	Ord (streng).
Output	
`N`	Antal stavelser (heltal).

Eksempel

Forestil dig input-ordet `cheesecake`. Ordet indeholder en dobbelt-vokal samt tre enkelt-vokaler hvoraf én er i slutningen af ordet.

`ch`<u>`ee`</u>`s`<u>`e`</u>`c`<u>`a`</u>`ke`

Antallet af stavelser skal således estimeres til $N = 3$. De bidragende vokaler er uderstreget.