

Levenshtein-távolság

A következő algoritmus két, maximum 25 karakter hosszú karakterláncról megállapítja, hogy mekkora a Levenshtein-távolságuk, azaz minimálisan hány karakterenkénti művelet (beszúrás, törlés, csere) kell ahhoz, hogy az egyik karakterláncot a másikra átalakítsuk.

Kódolja az algoritmust függvényekbe! Az elkészült forráskódot *SajátNév.cs* néven mentse, s töltsse fel!

A megoldás során vegye figyelembe a következőket:

- *A "Hossz()" függvény a karakterlánc hosszát adja meg.*
- *Az „-ig” értelmezésénél legyen benne az érték is.*
- *A "Térj vissza!” utasítás megszakítja a függvény futását, és meghatározza annak visszatérési értékét.*

Függvény Min(a: Egész, b: Egész): Egész

Ha $a < b$ akkor

Min:= a

különben

Min:= b

Függvény vége

Függvény LDTav(s1: Szöveg, s2: Szöveg): Egész

Változó h1, h2:Egész

Változó c:Egész

Változó tömb m[0..26, 0..26]:Egész

h1:= Hossz(s1)

h2:= Hossz(s2)

Ha $h1 = 0$ akkor Térj vissza h2-vel

Ha $h2 = 0$ akkor Térj vissza h1-el

Ciklus i:= 0-tól h1-ig (+1 lépésközzel)

m[i,0]:= i

Ciklus vége

Ciklus j:= 0-tól h2-ig (+1 lépésközzel)

m[0,j]:= j

Ciklus vége

Ciklus i:= 1-től h1-ig (+1 lépésközzel)

Ciklus j:= 1-től h2-ig (+1 lépésközzel)

c:= 1

Ha $s2[j-1] = s1[i-1]$ akkor c:= 0

$m[i,j] := \text{Min}(\text{Min}(m[i-1,j]+1, m[i,j-1]+1), m[i-1,j-1]+c)$

Ciklus vége

Ciklus vége

Térj vissza m[h1,h2]-vel

Függvény vége

Függvény Main() Üres

Ki: LDTav("alma", "halmaz")

Függvény vége

Pontok

Ebben a feladatban egy szöveges állományban x, y koordinátákkal megadott pontokat kell vizsgálnia. Minden pont adata külön sorba került a forrásállományban a sorszámuk alapján növekvő rendben a következők szerint:

- A sorok a "P(" karakterlánccal kezdődnek.
- A következő 3 karakteren a pont sorszáma található jobbra igazítva.
- A 6-7. karakteren a ")=" karakterlánc található.
- 8-10. karakterek a x koordináta értékét tárolják ($-99 \leq x \leq 99$) jobbra igazítva.
- A 11. karakteren a vessző karakter található.
- A 12-14. karakterek az y koordináta értékét tárolják ($-99 \leq y \leq 99$) jobbra igazítva.

Például a következő sor a 77. pont koordinátáit adja meg:

Minta sor:	P	(7	7)	=		-	2	,			8
Karakter sorszáma	1	2	3	4	5	6	7	8	9	10	11	12	13	14

ahol: $x = -2$, $y = 8$

A megoldás során vegye figyelembe a következőket:

- Programját úgy készítse el, hogy tetszőleges adatok mellett is helyes eredményt adjon!
- A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat:)!
- A program megírásakor az adatok helyességét, érvényességét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.

Készítsen programot *pontok* néven, amely az alábbi feladatokat oldja meg! Az egyes feladatokban a kiírásokat a minta szerint készítse el! A megoldását CLI felületen készítse el, s a program forráskódját *pontok_SajátNév.cs* néven töltsse fel!

1. A program olvassa be az UTF-8 kódolású *pontok.txt* állományban lévő adatokat és tárolja el egy olyan adatszerkezetben, ami a további feladatok megoldására alkalmas! Jelenítse meg a képernyőn a *pontok.txt* állományban tárolt pontok számát!
2. A program határozza meg és írja ki, hogy hány pont található az x vagy az y tengelyen! (Egy pont akkor van az X tengelyen, ha az X koordinátája nulla, és akkor van az Y tengelyen, ha az Y koordinátája nulla.)
3. Keresse meg az azonos koordinátájú pontokat, majd írja ki a képernyőre a pontok koordinátáit és a pontok sorszámait a minta szerint! Ügyeljen arra, hogy minden koordináta 3 karakter széles legyen a kiírásakor, és legyenek jobbra igazítva!
4. Tekintse a pontokat szakaszok lehetséges végpontjainak! Határozza meg a leghosszabb szakasz hosszát és írja ki a képernyőre! Feltételezheti, hogy csak egy ilyen pontpáros van az adatok között.

A $P_1(x_1, y_1)$ és a $P_2(x_2, y_2)$ végpontok által meghatározott szakasz hosszát a $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ kifejezéssel tudja meghatározni.

5. Az előző feladatban meghatározott pontpárost írja ki a max_hossz.txt állományba a forrásállomány (pontok.txt) karakterpontos felépítése szerint!

Minta:

1. feladat: Pontok száma a pontok.txt állományban: 150 db
2. feladat: Pontok száma az x vagy y tengelyen: 8 db
3. feladat: Azonos koordinátájú pontok:
 - Az x= 20 y= 9 koordinátán: 8. 129.
 - Az x=-10 y=-15 koordinátán: 12. 118.
 - Az x= 13 y=-12 koordinátán: 22. 117.
 - Az x= 6 y= 6 koordinátán: 24. 37.
 - Az x=-20 y= 12 koordinátán: 35. 124.
 - Az x= 14 y= 11 koordinátán: 51. 90. 143.
4. feladat: Leghosszabb szakasz hossza: 54,4518135602479
5. feladat: max_hossz.txt