Hasonlóságelemzés macro-alapon

(Similarity analyses based on Excel-macros)

Pitlik László, Pitlik Marcell (MY-X team)

Kivonat: A programozás tanulása és tanítása végtelen sok módon történhet. Ezek hatásosságát és hatékonyságát általában véve nem szokás mérni. A programozásra való alkalmasság eleve elvár egy magas fokú szuverenitásra törekvést, az autodidakta tanulás iránti vágyat/készséget. Így vélelmezhetően minden olyan esettanulmány, mely hasonló/azonos funkcionalitások eltérő nyelveken, eltérő logikák mentén történő bemutatását teszi lehetővé lényegében teljes részletgazdagsággal, az a szuverén és autodidakta tanulást érdemben képes illene, hogy legyen támogatni. A cikk célja programozástanulást elszakítani a klasszikus programnyelv-tanulás/tanítás logikájától és elsőként/párhuzamosan arra felhívni a figyelmet, hogy a programozás filozófiája terén is illik fejlődni, nem csak az adott programnyelven belüli virtuóz megoldások mentén. A cikkben az MS Excel macro-támogatása kerül fókuszba laikus programozó által készítve a kódot.

Kulcsszavak: COCO, Solver

Abstract: The learning and teaching in case of creating of source codes can be realized/executed in quasi unlimited ways. The effectivity and/or efficiency of these ways will not be measured in general. The capability of creating source codes assumes (in particular) a kind of struggling toward sovereignty and the high acceptance/desire level of auto-didactical learning methods. Therefore, each deep detailed case study (source code) should be useful for auto-didactive learners/teachers where the same/similar functionalities will be covered through source codes using other programming languages and/or programming structures. The article tries to create significant distances between language-oriented learning/teaching logics and the planning of creating qualitative source codes in a quasi language-independent way in order to catalyse more philosophy behind the source codes. The article focuses on a MS-Excel-macro solution created by a not professional programmer.

Keywords: COCO, Solver

# Bevezetés

A hasonlóságelemzésre már számos megoldás született az elmúlt évtizedekben:

* <http://miau.gau.hu/lps/>
* <https://miau.my-x.hu/myx-free/>
  + <https://miau.my-x.hu/myx-free/coco/index.html>
  + <https://miau.my-x.hu/myx-free/index_e.php3?x=e01>
* <https://www.google.com/search?q=sealog+seacon&oq=sealog+seacon>
* <http://miau.gau.hu/mm/Putnoki_Gyula.ppt>

* <https://miau.my-x.hu/miau/255/coco_js.docx>
* (<https://miau.my-x.hu/miau/255/coco_macro.docx>)

# A kód

<https://miau.my-x.hu/digeco/coco_prg1.xlsm> (macro3)

Sub Makró3()

' Billentyűparancs: Ctrl+r

Rem it is necessary to declare the number of rows (rws) and columns (clmns) in advance

rws = Sheets("coco").Cells(1, 2)

clmns = Sheets("coco").Cells(2, 2)

Rem avoiding problems concerning the number of rows and columns

Rem estimation of the next free row (nfr) for the ranked OAM

nfr = 4 + rws + 10

Rem The constant of 10 means rows for directions, numbers, meanings, dimensions, etc.

Rem first raw value (grey zone) should always be the sheets("coco").cells(4,2)!!! (see constant of 4)

Sheets("coco").Cells(nfr, 1) = "ranked OAM"

For i = 1 To rws

Sheets("coco").Cells(nfr + i, 1) = Sheets("coco").Cells(4 + i - 1, 1)

Next i

For i = 1 To clmns

Sheets("coco").Cells(nfr, i + 1) = Sheets("coco").Cells(4 - 1, i + 1)

Next i

set\_27 = -(nfr + 1 - 4)

set4 = 4

set19 = set4 + rws - 1

set21 = set19 + 2

frmlstr = "=RANK(R[" & set\_27 & "]C,R4C:R" & set19 & "C,R" & set21 & "C)"

Rem the relative co-ordinates of the concerned cells produce the same effect like the hand-made copy+paste-action (see modul2 macro5)

Rem the set of [-27]\_4\_19\_21 is just valid for the case where rws=16 - therefore these values should be re-calculated for each rws-value!

For i = 1 To rws

For j = 1 To clmns

Rem Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = "=RANK(R[-27]C,R4C:R19C,R21C)" <--coming from module2 macro4 - from a hand-made macro being valid just for rws=16

Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = frmlstr

Next j

Next i

nfr = nfr + rws + 10

Rem the next free row for scrolling down should always be fine tuned

Sheets("coco").Cells(nfr, 1) = "stairs"

For i = 1 To rws

Sheets("coco").Cells(nfr + i, 1) = i

Next i

For i = 1 To clmns

Sheets("coco").Cells(nfr, i + 1) = Sheets("coco").Cells(4 - 1, i + 1)

Next i

For i = 1 To rws

For j = 1 To clmns

Sheets("coco").Cells(nfr + i, j + 1).Select

Application.Run "coco\_prg1.xlsm!Makró6"

Next j

Next i

For i = 1 To clmns

Sheets("coco").Cells(nfr + rws + 1, i + 1) = i + 1

Next i

Rem this is the return-value for the vlookup-function (see later)

nfr = nfr + rws + 10

Rem the next free row for scrolling down should always be fine tuned

Sheets("coco").Cells(nfr, 1) = "differences of stairs"

For i = 1 To rws - 1

Sheets("coco").Cells(nfr + i, 1) = "s" & i & "- s" & (i + 1)

Next i

Rem pairwise visualization of the stairs in the row-headers

For i = 1 To clmns

Sheets("coco").Cells(nfr, i + 1) = Sheets("coco").Cells(4 - 1, i + 1)

Next i

frmlstr = "=R[" & set\_27 + 1 & "]C-R[" & set\_27 + 2 & "]C"

Rem it is not necessary to derive new set\_variables because the difference of 1 and 2 can be handled based on the variable of set\_27

For i = 1 To rws - 1

For j = 1 To clmns

Rem Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = "=R[-26]C-R[-25]C"

Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = frmlstr

Next j

Next i

Rem the same problem about adaptation to the real rws-value like before (see set\_27, etc.)

nfr = nfr + rws + 10

Rem the next free row for scrolling down should always be fine tuned

Sheets("coco").Cells(nfr, 1) = "estimations"

For i = 1 To rws

Sheets("coco").Cells(nfr + i, 1) = i

Next i

For i = 1 To clmns + 1

Sheets("coco").Cells(nfr, i + 1) = Sheets("coco").Cells(4 - 1, i + 1)

Next i

For i = 1 To rws

Sheets("coco").Cells(nfr + i, clmns + 2) = Sheets("coco").Cells(4 + i - 1, clmns + 2)

Next i

set\_78 = -(3 \* (rws + 10))

set57 = 2 \* (rws + 10) + 1 + 4

set72 = set57 + rws - 1

set7 = clmns + 1

frmlstr = "=VLOOKUP(R[" & set\_78 & "]C,R" & set57 & "C1:R" & set72 & "C" & set7 & ",R" & set72 + 1 & "C,0)"

Rem it is not necessary to derive new set\_variables

For i = 1 To rws

For j = 1 To clmns

Rem Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = "=VLOOKUP(R[-78]C,R57C1:R72C7,R73C,0)"

Sheets("coco").Cells(nfr + i, j + 1).FormulaR1C1 = frmlstr

Next j

Next i

Rem the same problem about adaptation to the real rws-value like before (see set\_27, etc.)

Sheets("coco").Cells(nfr, clmns + 3) = "sum"

For i = 1 To rws

Sheets("coco").Cells(nfr + i, clmns + 3).FormulaR1C1 = "=SUM(RC[-7]:RC[-2])"

Next i

Sheets("coco").Cells(nfr, clmns + 4) = "difference"

For i = 1 To rws

Sheets("coco").Cells(nfr + i, clmns + 4).FormulaR1C1 = "=RC[-1]-RC[-2]"

Next i

Sheets("coco").Cells(nfr + rws + 1, clmns + 4) = "error"

set\_back = -(rws + 1)

frmlstr = "=SUMPRODUCT(R[" & set\_back & "]C:R[-2]C,R[" & set\_back & "]C:R[-2]C)"

Sheets("coco").Cells(nfr + rws + 2, clmns + 4).FormulaR1C1 = frmlstr

Sheets("coco").Cells(nfr + rws + 2, clmns + 4).Select

trgt\_rw = nfr + rws + 2

trgt\_clmn = clmns + 4

GoSub 111

code = trgt\_str & trgt\_rw

trgt\_clmn = clmns + 1

GoSub 111

code2 = "B" & set57 & ":" & trgt\_str & set72

set83 = set57 + rws + 10

set97 = set83 + rws - 2

code3 = "B" & set83 & ":" & trgt\_str & set97

Rem before calling solver: tools / references / solver AFTER activating solver it is also necessary to reset the paramters!

SolverReset

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverAdd CellRef:=Range(code3), Relation:=3, FormulaText:="1"

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverSolve

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverSolve

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverSolve

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverOk SetCell:=Range(code), MaxMinVal:=2, ValueOf:=0, ByChange:=Range(code2), \_

Engine:=1, EngineDesc:="GRG Nonlinear"

SolverSolve

Stop

111 Rem conversions between numbers and letter for column-ids

If trgt\_clmn = 1 Then trgt\_str = "A"

If trgt\_clmn = 2 Then trgt\_str = "B"

If trgt\_clmn = 3 Then trgt\_str = "C"

If trgt\_clmn = 4 Then trgt\_str = "D"

If trgt\_clmn = 5 Then trgt\_str = "E"

If trgt\_clmn = 6 Then trgt\_str = "F"

If trgt\_clmn = 7 Then trgt\_str = "G"

If trgt\_clmn = 8 Then trgt\_str = "H"

If trgt\_clmn = 9 Then trgt\_str = "I"

If trgt\_clmn = 10 Then trgt\_str = "J"

If trgt\_clmn = 11 Then trgt\_str = "K"

If trgt\_clmn = 12 Then trgt\_str = "L"

If trgt\_clmn = 13 Then trgt\_str = "M"

If trgt\_clmn = 14 Then trgt\_str = "N"

If trgt\_clmn = 15 Then trgt\_str = "O"

If trgt\_clmn = 16 Then trgt\_str = "P"

If trgt\_clmn = 17 Then trgt\_str = "Q"

If trgt\_clmn = 18 Then trgt\_str = "R"

If trgt\_clmn = 19 Then trgt\_str = "S"

If trgt\_clmn = 20 Then trgt\_str = "T"

If trgt\_clmn = 21 Then trgt\_str = "U"

If trgt\_clmn = 22 Then trgt\_str = "V"

If trgt\_clmn = 23 Then trgt\_str = "W"

If trgt\_clmn = 24 Then trgt\_str = "X"

If trgt\_clmn = 25 Then trgt\_str = "Y"

If trgt\_clmn = 26 Then trgt\_str = "Z"

Return

End Sub

# Konklúziók

A megoldások közötti alapvető különbségek a tényleges funkcionalitásokon túl (pl. rangsorolás része, vagy nem része a folyamatnak) a programozás során felvállalt gondolati komplexitásban érhető tetten: vö.

* <https://miau.my-x.hu/miau/255/macro_tanulas_tanitas.docx>
* <https://miau.my-x.hu/miau/254/SUDOKU_SOLVER_CUSTOMIZED_SOURCE_CODE.docx>
* <https://miau.my-x.hu/miau/254/cipher1-2-3.docx>

A kódírás előtt tehát érdemes/illik szinte tetszőleges részletességű terveket készíteni az egyes döntések miértjeinek tételes átgondolása/validálása mellett.