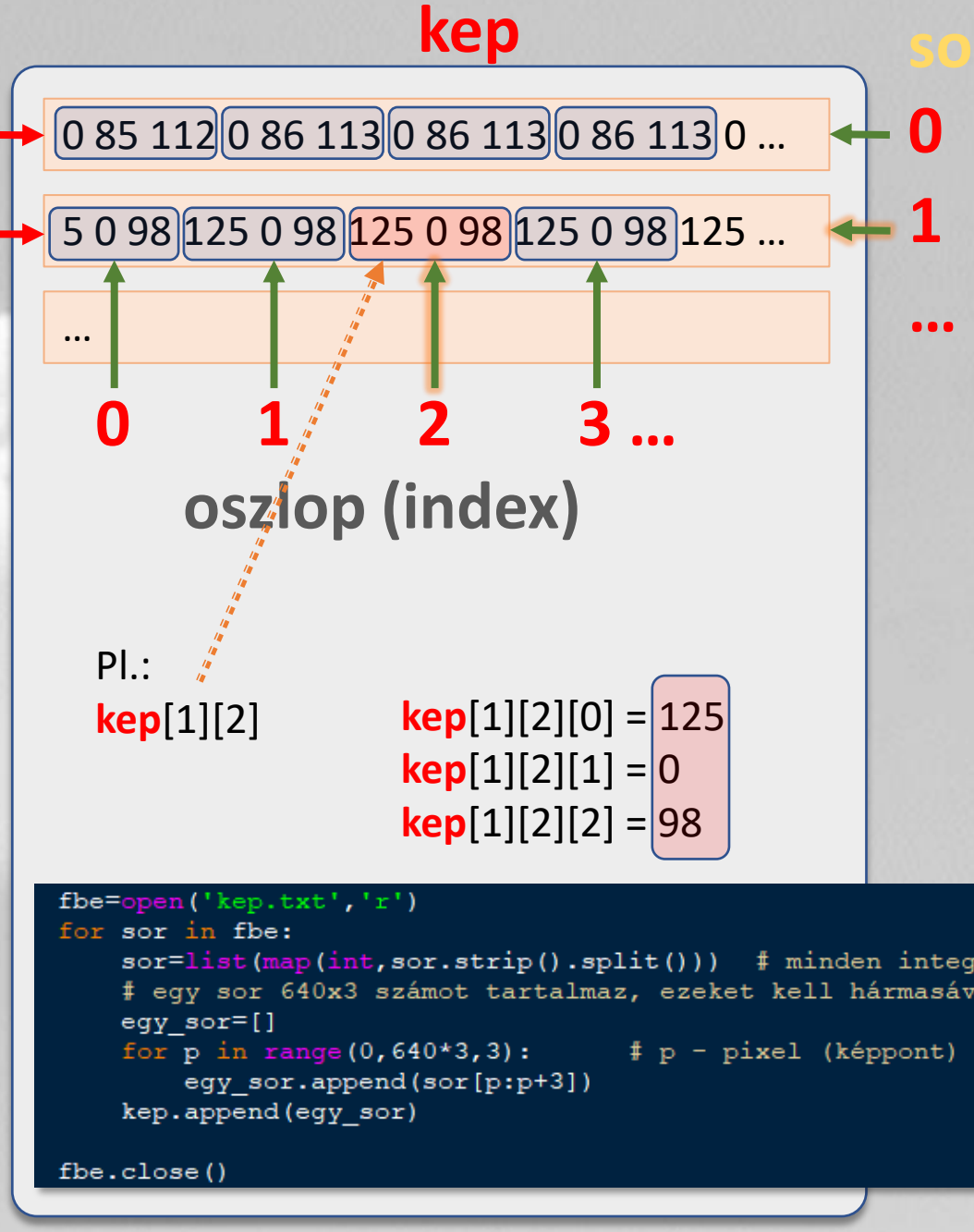
A forrásfájlban található adatok értelmezése ⟶ **mátrix** (**3D**) - „listák listájának listája” – egy táblázat, ahol a táblázat elemei *nem egyedi értékek, hanem listák* (3 elemmel)

kep.txt – Jegyzettömb

Fájl Szerkesztés Formátum Nézet Súgó

```
0 85 112 0 86 113 0 86 113 0 86 113 0
5 0 98 125 0 98 125 0 98 125 0 98 125
3 130 4 104 130 5 105 131 5 105 131 5
8 133 14 108 133 14 108 133 17 109 134
136 17 111 136 17 111 136 17 111 136 1
35 20 112 135 20 112 135 20 112 135 26
3 16 108 133 14 108 133 14 108 133 14
0 86 113 0 86 113 0 86 113 0 86 113 0
5 0 98 125 0 98 125 0 98 125 0 98 125
103 130 4 104 130 5 105 131 5 105 131
108 133 14 108 133 14 108 133 17 109 1
11 136 17 111 136 17 111 136 17 111 13
2 135 20 112 135 20 112 135 20 112 135
133 16 108 133 14 108 133 14 108 133 1
0 86 111 0 86 111 0 86 113 0 86 113 0
5 0 98 125 0 98 125 0 98 125 0 98 125
103 130 4 104 130 5 105 131 5 105 131
108 133 14 108 133 14 108 133 17 109 1
11 136 17 111 136 17 111 136 17 111 13
2 135 20 112 135 20 112 135 20 112 135
133 16 108 133 16 108 133 16 108 133 1
0 86 113 0 86 113 0 86 113 0 86 113 1
5 0 98 125 0 98 125 0 98 125 0 98 125
103 130 4 104 130 5 105 131 5 105 131
108 133 15 109 134 15 109 134 17 109 1
12 137 18 112 137 18 112 137 18 112 13
2 135 20 112 135 20 112 135 20 112 135
```




```python
fbe=open('kep.txt','r')
for sor in fbe:
    sor=list(map(int,sor.strip().split()))  # minden integer (egész szám) legyen
    # egy sor 640x3 számot tartalmaz, ezeket kell hármasával "csoportosítani"
    egy_sor=[]
    for p in range(0,640*3,3):      # p - pixel (képpont)
        egy_sor.append(sor[p:p+3])
    kep.append(egy_sor)

fbe.close()
```

rgb202305.py - C:\Users\admin\Documents\202305emelt_informatika\rgb202305.py (3.11.1)

File Edit Format Run Options Window Help

```python
kep=[]          # ez lesz a 3 dimenziós mátrix
                # ---> egy képpont elérése: kep[i][j]---> [r,g,b]
                # ---> azon belül R=kep[i][j][0], G=kep[i][j][1], B=kep[i][j][2]

def F1():
    fbe=open('kep.txt','r')
    #mátrix felépítése: olyan lista, melynek elemei LISTÁKBÓL álló listák
    for sor in fbe:
        sor=list(map(int,sor.strip().split()))  # minden integer (egész szám) le
        # egy sor 640x3 számot tartalmaz, ezeket kell hármasával "csoportosítani
        egy_sor=[]
        for p in range(0,640*3,3):      # p - pixel (képpont)
            egy_sor.append(sor[p:p+3])
        kep.append(egy_sor)

    fbe.close()
    # print(kep) - ennek a végrehajtása több, mint egy percig tartott a gépemen,
    # Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 16GB RAM
    # utána az ablak bezárása majdnem kifagyáshoz vezetett, de aztán mégsem... :

def F2():
    print('2. feladat:')
    print('Kérem egy képpont adatait!')
    sor=int(input('Sor: '))-1                   # 180    -1 az indexelés mia
    oszlop=int(input('Oszlop: '))-1             # 320
    RGB=kep[sor][oszlop]    # az adott sor és oszlop indexen egy 3 elemű lista v
    R=RGB[0]
    G=RGB[1]
    B=RGB[2]
    print(f'A képpont színe RGB({R},{G},{B})')      # RGB(184,183,181)

def F3():
    print('3. feladat:')
    # világos képpont: RGB-értékeinek összege 600-nál nagyobb
    vk=0
    # kep mátrix 2D bejárása, az elemei 3 elemű listák, ugye:
    for kepsor in kep:
        for keppont in kepsor:
            if sum(keppont)>600:            # a listára alkalmazható a sum függvé
                vk+=1
    print(f'A világos képpontok száma: {vk}')       # 7837

def kulcsfuggveny(keppont):     # rendezéses minimumkereséshez kell...
    return sum(keppont)

def F4():
```

Ln: 43 Col: 70

rgb202305_class_temp.py - C:/Users/admin/Documents/202305emelt_informatika/rgb202305_class_temp.py (3.11.1)

File Edit Format Run Options Window Help

```python
# a keppont osztály egy listából "építkezik" (lst),
# mely egy RGB számhármas (lsd. 32. sor!)
class keppont:
    def __init__(self,lst):
        self.r=lst[0]
        self.g=lst[1]
        self.b=lst[2]

    def osszeg(self):
        return self.r+self.g+self.b

    def vilagos(self):
        if keppont.osszeg(self)>600:
            return True
        return False
#-----------------------------------------------------------------------------
# JELÖLÉS:  "ko" az alábbi részekben egy keppont class-ba tartozó objektumot fog
#           (röviden: "ko" egy keppont objektum)
#-----------------------------------------------------------------------------

kep=[]          # ez lesz a 2 dimenziós mátrix
                # ---> egy képpont elérése: kep[i][j]---> keppont objektum
                # ---> azon belül R=kep[i][j].r, G=kep[i][j].g, B=kep[i][j].b

def F1():
    fbe=open('kep.txt','r')
    # mátrix felépítése: olyan lista, melynek elemei OBJEKTUMOKBÓL álló listák
    for sor in fbe:
        sor=list(map(int,sor.strip().split()))  # minden integer (egész szám) leg
        # egy sor 640x3 számot tartalmaz, ezeket kell hármasával "csoportosítani"
        egy_sor=[]
        for p in range(0,640*3,3):      # p - pixel (képpont)
            egy_sor.append(keppont(sor[p:p+3]))     # itt "generálom" a keppont o
        kep.append(egy_sor)                         # egy_sor egy ko (objektum) l

    fbe.close()
    # print(kep) - ennek a végrehajtása több, mint egy percig tartott a gépemen,
    # Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 16GB RAM
    # utána az ablak bezárása majdnem kifagyáshoz vezetett, de aztán mégsem... :)

def F2():
    print('2. feladat:')
    print('Kérem egy képpont adatait!')
    sor=int(input('Sor: '))-1                   # 180    -1 az indexelés miat
    oszlop=int(input('Oszlop: '))-1             # 320
    ko=kep[sor][oszlop]     # az adott sor és oszlop indexen egy keppont objektum
```



Ln: 56 Col: 0

Írjon ide a kereséshez
Asztal
26°C Napos
19:20
2023.05.23.

rgb202305.py - C:\Users\admin\Documents\202305emelt_informatika\rgb202305.py (3.11.1)

File Edit Format Run Options Window Help

```python
kep=[]          # ez lesz a 3 dimenziós mátrix
                # ---> egy képpont elérése: kep[i][j]---> [r,g,b]
                # ---> azon belül R=kep[i][j][0], G=kep[i][j][1], B=kep[i][j][2]

def F1():
    fbe=open('kep.txt','r')
    #mátrix felépítése: olyan lista, melynek elemei LISTÁKBÓL álló listák
    for sor in fbe:
        sor=list(map(int,sor.strip().split()))  # minden integer (egész szám) le
        # egy sor 640x3 számot tartalmaz, ezeket kell hármasával "csoportosítani
        egy_sor=[]
        for p in range(0,640*3,3):      # p - pixel (képpont)
            egy_sor.append(sor[p:p+3])
        kep.append(egy_sor)

    fbe.close()
    # print(kep) - ennek a végrehajtása több, mint egy percig tartott a gépemen,
    # Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 16GB RAM
    # utána az ablak bezárása majdnem kifagyáshoz vezetett, de aztán mégsem... :

def F2():
    print('2. feladat:')
    print('Kérem egy képpont adatait!')
    sor=int(input('Sor: '))-1                   # 180    -1 az indexelés mia
    oszlop=int(input('Oszlop: '))-1             # 320
    RGB=kep[sor][oszlop]    # az adott sor és oszlop indexen egy 3 elemű lista v
    R=RGB[0]
    G=RGB[1]
    B=RGB[2]
    print(f'A képpont színe RGB({R},{G},{B})')      # RGB(184,183,181)

def F3():
    print('3. feladat:')
    # világos képpont: RGB-értékeinek összege 600-nál nagyobb
    vk=0
    # kep mátrix 2D bejárása, az elemei 3 elemű listák, ugye:
    for kepsor in kep:
        for keppont in kepsor:
            if sum(keppont)>600:        # a listára alkalmazható a sum függvé
                vk+=1
    print(f'A világos képpontok száma: {vk}')       # 7837

def kulcsfuggveny(keppont):     # rendezéses minimumkereséshez kell...
    return sum(keppont)

def F4():
```

Ln: 43 Col: 70

rgb202305_class_temp.py - C:/Users/admin/Documents/202305emelt_informatika/rgb202305_class_temp.py (3.11.1)

File Edit Format Run Options Window Help

```python
kep=[]          # ez lesz a 2 dimenziós mátrix
                # ---> egy képpont elérése: kep[i][j]---> keppont objektum
                # ---> azon belül R=kep[i][j].r, G=kep[i][j].g, B=kep[i][j].b

def F1():
    fbe=open('kep.txt','r')
    # mátrix felépítése: olyan lista, melynek elemei OBJEKTUMOKBÓL álló listák
    for sor in fbe:
        sor=list(map(int,sor.strip().split()))  # minden integer (egész szám) leg
        # egy sor 640x3 számot tartalmaz, ezeket kell hármasával "csoportosítani"
        egy_sor=[]
        for p in range(0,640*3,3):      # p - pixel (képpont)
            egy_sor.append(keppont(sor[p:p+3]))     # itt "generálom" a keppont o
        kep.append(egy_sor)                         # egy_sor egy ko (objektum) l

    fbe.close()
    # print(kep) - ennek a végrehajtása több, mint egy percig tartott a gépemen,
    # Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 16GB RAM
    # utána az ablak bezárása majdnem kifagyáshoz vezetett, de aztán mégsem... :)

def F2():
    print('2. feladat:')
    print('Kérem egy képpont adatait!')
    sor=int(input('Sor: '))-1                   # 180    -1 az indexelés miat
    oszlop=int(input('Oszlop: '))-1             # 320
    ko=kep[sor][oszlop]     # az adott sor és oszlop indexen egy keppont objektum
    R=ko.r
    G=ko.g
    B=ko.b
    print(f'A képpont színe RGB({R},{G},{B})')      # RGB(184,183,181)

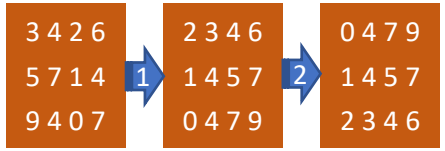
def F3():
    print('3. feladat:')
    # világos képpont: RGB-értékeinek összege 600-nál nagyobb
    vk=0
    # kep mátrix 2D bejárása, az elemei keppont objektumok, ugye:
    for kepsor in kep:
        for ko in kepsor:
            if ko.vilagos():        # alkalmazható az objektum "vilagos" metód
                vk+=1
    print(f'A világos képpontok száma: {vk}')       # 7837

def kulcsfuggveny(ko):
    return ko.osszeg()

def F4():
```

Ln: 76 Col: 25

Írjon ide a kereséshez
Asztal
26°C Napos
19:22
2023.05.23.



kep *minden* listájában a ***0. indexen*** a **legkisebb** RGB összegű képpont lesz!

2. **kep** listáinak átrendezése a ***0. indexű elemek*** alapján!

```
[sorted(kepsor,key=kulcsfuggveny) for kepsor in kep]
```

1. **kep** egy listáját rendezi **RGB összeg alapján** növekvő sorrendbe, és ezt a **kep *minden* listájával „elköveti”**!

```
63 def kulcsfuggveny(ko):
64     return ko.osszeg()
65
66 def F4():
67     print('4. feladat')
68     # ELSŐ KÖRBEN minimumkeresés az RGB összegek között rendezéssel
69     rendezett=sorted([sorted(kepsor,key=kulcsfuggveny) for kepsor in kep],key=lambda x: kulcsfuggveny(x[0]))
70     minimum=rendezett[0][0].osszeg()
71     print(f'A legsötétebb pont RGB összege: {minimum}')      # 197
72
73     # MÁSODIK KÖRBEN szedjük össze a minimummal egyező képpontokat:
74     minimumok=[]    # egy biztos lesz...
75     for kepsor in kep:
76         for ko in kepsor:
77             if ko.osszeg()==minimum:
78                 minimumok.append(ko)    # minimumok listába beletesszük a mátrix elemét, a ko-t
79
80     print('A legsötétebb pixelek színe:')
81     for ko in minimumok:        # minimumok egy ko-kat tartalmazó lista
82         print(f'RGB({ko.r},{ko.g},{ko.b})')          # RGB(0,85,112)  RGB(0,86,111)  RGB(0,86,111)
```

3426
5714
9427