

## Default érték megadása

```
from __future__ import print_function

def info(net, diameter=False, welcome="Hello"):
    print(welcome)
    print("N =", net.vcount())
    print("M =", net.ecount())
    if diameter:
        print("d =", net.diameter())
    return net.vcount()

import igraph
fullnet = igraph.Graph.Full(8)
info(fullnet)
info(fullnet, True)
info(fullnet, welcome="Ciao!") # Kulcsszó nélküli elől.
info(welcome="Szia!", net=fullnet, diameter=True)
```

## Metódusok írása hálózathoz

maxmindeg.py

```
import igraph
class Network(igraph.Graph):
    def maxmindegree(self, write=True):
        "Returns with the maximal and minimal degree."
        deg = self.degree()
        if write:
            print("Maximal degree = {0}, "
                  "minimal degree = {1}".format(
                      max(deg),min(deg)))
        return max(deg), min(deg)

net = Network.Full(8)
maxd, mind = net.maxmindegree(write=False)
print(net.maxmindegree())
# Maximal degree = 7, minimal degree = 7
print(net.maxmindegree.__doc__)
# Returns with the maximal and minimal degree.
```

## unittest

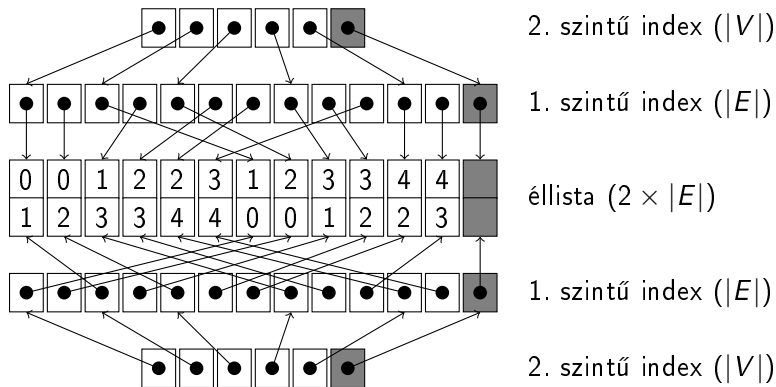
test\_maxmindeg.py

```
import maxmindeg
import unittest

class testNetwork(unittest.TestCase):
    def testDegree(self):
        net = maxmindeg.Network.Full(8)
        maxd, mind = net.maxmindegree(write=False)
        self.assertEqual(maxd, 7)
        self.assertEqual(mind, 7)

if __name__ == "__main__":
    unittest.main()
```

# Éllista kettős indexeléssel (igraph)



Összes tárigény:  $4|E| + 2|V|$

