

Based on Python recursion

- assumes that each network feature has a "downId" value (see [these pages for guidance on that](#)).
- assumes that entire network can be dumped to an array in Python.
- does not assume any proprietary GIS platform
- just an enhanced version of [Finding upstream flowlines - recursion \(Python\)](#)

This pattern could be replicated in any language, but is implemented here in Python. It is a tabular manipulation, requiring no GIS software, but relies on the **id** and **idDown** attributes to define routing connectivity.

The hydrographic network is known to the function as an array, **networkArr**, where each record has two attributes: 1) the feature identifier, **id**, and 2) the identifier of the downstream feature, **downId**. The array is "typed", meaning that the columns have names (strings).

If there are any features whose **downId** points to the **id**, then the function will *call itself* once for each inflowing neighbor using that neighbor's **id** as the **id** argument. If there is a feature draining into one of the inflowing neighbors, then the function will call itself using *that neighbor's id*. This process will climb upslope according to the **networkArr** connectivity, until the **networkArr** indicates that no records have an **idDown** value that equals the current **id** (i.e., that the current flowline feature is a headwater). The final product of the function is the **networkArr**, but that has updated attributes. Note that the initial version of **networkArr** should provide an accumulation field (**charAccumField**) that is initialized to the **charField** on the first iteration.

Note that this sample does not include any code to write results to a file or back to a GIS format. Pretty basic, but you'll still need to deal with that (at least until this or a link page gets updated).

```
def accumCharUpstream(networkArr, id, idField, idDownField, charField, charAccumField):
    inFlowlines = networkArr[networkArr[idDownField] == id]
    charVal = networkArr[charAccumField][networkArr[idField] == id]
    try:
        for inFlowline in inFlowlines:
            upId = inFlowline[idField]
            networkArr[charAccumField][networkArr[idField] == upId] += charVal
            networkArr = accumCharUpstream(networkArr, upId, idField,
                                           idDownField,
                                           charField, charAccumField)
    except:
        pass
    return networkArr
```

Below is an example of some **arcpy** code for extracting the relevant information from a featureclass, as well as the invocation of the accumulation method. Note that the field to be accumulated ('SHAPE@LENGTH') is arbitrary.

```
import arcpy
networkArr = arcpy.da.TableToNumPyArray(myNetworkFeatureClass,
                                         ['COMID', 'idDown', 'SHAPE@LENGTH'])

networkArr = ...some garbage to add rename 'SHAPE@LENGTH' column to just 'length' and add 'lenTotal' column

# initialize accumulation field, 'lenTotal'
fromTo['lenTotal'] = fromTo['length']

# run the function once for each feature that drains to nobody (assumed to be an outlet)
for id in networkArr['COMID'][networkArr['downId'] == 0]:
    fromTo = accumCharUpstream(networkArr, id, 'COMID', 'idDown', 'length', 'lenTotal')
```

User	Edits	Comments	Labels
Viger, Roland	6	1	0
Wieczorek, Michael E.	0	1	0