

```

import numpy as np

# Forward star representation [T, H, C, U]
# [0, 0, 0] is a dummy node.
FStar = np.array([[0, 0, 0],
                  [1, 2, 2],
                  [1, 3, 8],
                  [2, 3, 5],
                  [2, 4, 3],
                  [3, 2, 6],
                  [3, 5, 0],
                  [4, 3, 1],
                  [4, 5, 7],
                  [4, 6, 6],
                  [5, 4, 4],
                  [6, 5, 2]])

def adjacency_list(FStar):
    """
    Outarc adjacency list
    """
    A = dict()
    # Excludes the dummy node 0.
    N = np.sort(np.unique(FStar[:,0:2].reshape(-1,)))[1:]
    for i in N:
        A[i] = FStar[:,1][np.argwhere(FStar[:,0]==i)].reshape(-1).tolist()
    return A

def Dijkstra(FStar, s):
    """
    Dijkstra's algorithm
    """
    A = adjacency_list(FStar)
    n = len(A)
    S = list()
    S_bar = A.keys()
    d = np.ndarray((n+1,))
    d[:] = np.inf
    d[s] = 0
    d_bar = np.array(d)
    pred = np.zeros(n+1, dtype=int)
    pred[s] = 0
    while len(S) < n:
        i = np.argmin(d_bar)
        d_bar[i] = np.inf
        S.append(i)
        S_bar.remove(i)
        for j in A[i]:
            c = FStar[np.all(FStar[:,0:2]==[i,j], axis=1),2]
            if d[j] > d[i] + c:
                d[j] = d[i] + c
                d_bar[j] = d[j]
                pred[j] = i
    return d, pred

if __name__=="__main__":
    d, pred = Dijkstra(FStar, 1)
    print "Distance vector, d:"
    print d[1:]
    print "Pred vector, pred:"
    print pred[1:]

```