

Skrypt (Python 3.0):

```
import networkx as nx

D = nx.DiGraph() # tworzymy graf skierowany, na razie bez wierzchołków i bez krawędzi

D.add_nodes_from(['S', 1, 2, 'T']) # definiujemy zbiór wierzchołków

D.add_weighted_edges_from([ # definiujemy zbiór strzałek i ich pojemności
    ('S', 1, 4), # stworzyliśmy strzałkę (S,1) z pojemnością 4
    ('S', 2, 5),
    (1, 'T', 3),
    (2, 'T', 6),
], weight='capacity')

print(nx.maximum_flow(D, 'S', 'T')) # podaje przykładowy największy przepływ
print(nx.minimum_cut(D, 'S', 'T')) # podaje przykładowe najmniejsze cięcie
```

Wynik działania powyższego skryptu:

```
(8, {'S': {1: 3, 2: 5}, 1: {'T': 3}, 2: {'T': 5}, 'T': {}})
(8, ({1, 'S'}, {2, 'T'}))
```

Wyjaśnienie wyniku:

```
(8, {'S': {1: 3, 2: 5}, 1: {'T': 3}, 2: {'T': 5}, 'T': {}})
```

8 to wartość największego przepływu. Dalej czytamy wartości przepływu na strzałkach:
strzałce S->1 przypisano 3
strzałce S->2 przypisano 5
strzałce 1->T przypisano 3
strzałce 2->T przypisano 5
z T nie wychodzi strzałka, więc mamy pusty zbiór {} następników wierzchołka T.

```
(8, ({1, 'S'}, {2, 'T'}))
```

8 to pojemność najmniejszego cięcia
{1, 'S'} jest zbiorem wierzchołków po stronie źródła
{2, 'T'} jest zbiorem wierzchołków po stronie ujścia