

## Warunki

1. Zadania można wysyłać, jeśli ktoś ma najwyżej 2 plusy (licząc od 18 III).
2. W implementacjach można korzystać tylko z elementarnych konstrukcji Python'a (funkcje, instrukcje warunkowe, pętle, `range`, klasy użyte do definiowania struktur danych, wbudowana funkcja sortująca, itp.). **Nie wolno korzystać ze słowników i zbiorów.**
3. Rozwiązania muszą być efektywne obliczeniowo (także w zadaniach, w których nie podajemy wprost ograniczenia na złożoność obliczeniową). Zadania o zbyt wysokiej złożoności będą oceniane na brak plusa.
4. **Wolno omawiać zadania (w tym pomysły na implementację), ale wyłącznie na forum w systemie UPEL. Nie wolno wymieniać kodu realizującego fragmenty algorytmu (ale wolno odpowiadać na pytania postaci "jak zrealizować tablicę dwuwymiarową" itp.).**

## Zadanie 1 (maksymalny przepływ)

Dany jest graf skierowany  $G = (V, E)$  reprezentowany przez funkcję przepustowości  $c: V \times V \rightarrow \mathbb{N}$  oraz wierzchołki  $s$  i  $t$ . Proszę zaimplementować funkcję `max_flow( c, s, t )`, która zwraca wartość maksymalnego przepływu z wierzchołka  $s$  do  $t$ .

Państwa kod powinien mieć następującą postać (będzie uruchamiany; proszę nie usuwać fragmentu testującego; sprawdzający może także dołożyć swoje testy):

```
def max_flow( c, s, t ):
    # policz maksymalny przepływ z s do t
    # c[i][j] to przepustowość krawędzi z i do j
    # jeśli c[i][j] > 0 to c[j][i] = 0

    c = [[0 for j in range(4)] for i in range(4)]
    c[0][1] = 2
    c[0][2] = 1
    c[1][2] = 1
    c[1][3] = 1
    c[2][3] = 2
    print( max_flow( c, 0, 3 ) ) # wypisze 3
```