

Algorytmy i Struktury Danych  
Zadanie offline 9 (30.V.2022)

### Format rozwiązań

Rozwiązanie zadania musi się składać z **krótkiego** opisu algorytmu (wraz z uzasadnieniem poprawności) oraz jego implementacji. Zarówno opis algorytmu jak i implementacja powinny się znajdować w tym samym pliku Pythona (rozszerzenie `.py`). Opis powinien być na początku pliku w formie komentarza (w pierwszej linii w komentarzu powinno być imię i nazwisko studenta). Opis nie musi być długi—wystarczy kilka zdań, jasno opisujących ideę algorytmu. Implementacja musi być zgodna z szablonem kodu źródłowego dostarczonym wraz z zadaniem. Niedopuszczalne jest w szczególności:

1. korzystanie z zaawansowanych struktur danych (np. słowników czy zbiorów),
2. zmienianie nazwy funkcji implementującej algorytm, listy jej argumentów, lub nazwy pliku z rozwiązaniem,
3. modyfikowanie testów dostarczonych wraz z szablonem,
4. wypisywanie na ekranie jakichkolwiek napisów innych niż wypisywane przez dostarczony kod (ew. napisy dodane na potrzeby diagnozowania błędów należy usunąć przed wysłaniem zadania).

Dopuszczalne jest natomiast:

1. korzystanie z następujących elementarnych struktur danych: krotka, lista, kolejka `collections.deque`, kolejka priorytetowa (`queue.PriorityQueue` lub `heapq`),
2. korzystanie ze struktur danych dostarczonych razem z zadaniem (jeśli takie są).
3. korzystanie z wbudowanych funkcji sortujących (można założyć, że mają złożoność  $O(n \log n)$ ).

Wszystkie inne algorytmy lub struktury danych wymagają implementacji przez studenta. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych w pliku z szablonem rozwiązania.

Zadania niezgodne z powyższymi ograniczeniami otrzymają ocenę 0 punktów. Rozwiązania w innych formatach (np. `.PDF`, `.DOC`, `.PNG`, `.JPG`) z definicji nie będą sprawdzane i otrzymają ocenę 0 punktów, nawet jeśli będą poprawne.

### Testowanie rozwiązań

Żeby przetestować rozwiązanie zadania należy wykonać polecenie: `python3 zad9.py`

## Zadanie offline 9.

Szablon rozwiązania: zad9.py

W pewnym państwie znajdują się miasta, połączone siecią jednokierunkowych rurociągów, każdy o określonej przepustowości. Złoża ropy zostały wyczerpane, jednak w jednym z miast odkryto niewyczerpane źródło nowego rodzaju paliwa. Postanowiono zbudować dwie fabryki w różnych miastach oczyszczające nowe paliwo. Z pewnych względów fabryki te nie mogą znajdować się w mieście, w którym odkryto nowe złoża i nowe paliwo będzie transportowane istniejącą siecią rurociągów. Należy wskazać dwa miasta w których należy zbudować fabryki aby zmaksymalizować produkcję oczyszczonego paliwa.

Proszę zaimplementować funkcję `maxflow(G, s)`, która dla istniejącej sieci rurociągów `G` i miasta, w którym odkryto złożę `s`, zwróci maksymalną łączną przepustowość do dwóch miast w których należy zbudować fabryki. Miasta są ponumerowane kolejnymi liczbami  $0, 1, 2, \dots$ . Sieć rurociągów opisuje lista trójek: (miasto w którym rozpoczyna się rurociąg, miasto w którym się kończy rurociąg, przepustowość rurociągu)

**Przykład** Dla sieci  $G = [(0, 1, 7), (0, 3, 3), (1, 3, 4), (1, 4, 6), (2, 0, 9), (2, 3, 7), (2, 5, 9), (3, 4, 9), (3, 6, 2), (5, 3, 3), (5, 6, 4), (6, 4, 8)]$  oraz miasta  $s=2$  wynikiem jest 25 (miasta 4 i 5).