

## Algorytmy i Struktury Danych

### Zadanie offline 8 (23.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 zad8.py`

## Zadanie offline 8.

Szablon rozwiązania: zad8.py

W pewnym państwie, w którym znajduje się  $N$  miast, postanowiono połączyć wszystkie miasta siecią autostrad, tak aby możliwe było dotarcie autostradą do każdego miasta. Ponieważ kontynent, na którym leży państwo jest płaski położenie każdego z miast opisują dwie liczby  $x, y$ , a odległość w linii prostej pomiędzy miastami liczona w kilometrach wyraża się wzorem  $len = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ . Z uwagi na oszczędności materiałów autostrada łączy dwa miasta w linii prostej.

Ponieważ zbliżają się wybory prezydenta, wszystkie autostrady zaczęto budować równocześnie i jako cel postanowiono zminimalizować czas pomiędzy otwarciem pierwszej i ostatniej autostrady. Czas budowy autostrady wyrażony w dniach wynosi  $\lceil len \rceil$  (sufit z długości autostrady wyrażonej w km).

Proszę zaimplementować funkcję `highway(A)`, która dla danych położzeń miast wyznacza minimalną liczbę dni dzielącą otwarcie pierwszej i ostatniej autostrady.

**Przykład** Dla tablicy  $A = [(10, 10), (15, 25), (20, 20), (30, 40)]$  wynikiem jest 7 (Autostrady pomiędzy miastami 0-1, 0-2, 1-3).