

Warunki

1. W implementacjach można korzystać tylko z elementarnych konstrukcji Python'a (funkcje, instrukcje warunkowe, pętle, `range`, klasy użyte do definiowania struktur danych). **Nie wolno korzystać ze słowników i zbiorów, wbudowanego sortowania itp.**
2. Rozwiązania muszą być efektywne obliczeniowo (także w zadaniach, w których nie podajemy wprost ograniczenia na złożoność obliczeniową). Zadania o zbyt wysokiej złożoności będą otrzymywały obniżone oceny (lub 0).
3. Rozwiązania zadań proszę umieszczać w załączonych plikach (`zadX.py`).

Zadanie 1 (QuickSort na listach)

Proszę zaimplementować funkcję `qsort(L)` sortującą zadaną listę jednokierunkową (bez wartownika) przechowującą liczby. List reprezentowana jest przez węzły:

```
class Node:
    def __init__(self):
        self.next = Node
        self.value = None
```

Sortowanie powinno działać przez przepinanie węzłów. Proszę zwrócić uwagę, że sortowanie list przy pomocy QuickSort'a nie powinno używać funkcji `partition` z wykładu, ale odpowiednio zaadaptowaną funkcję, którą się wygodnie implementuje na listach (w szczególności wygodnie jest zaimplementować wariant QuickerSort, który dzieli listę na elementy mniejsze od pivotu, równe mu, oraz większe).