

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 (sortowanie przez scalanie)

W algorytmie sortowania przez scalanie mamy daną tablicę T i działamy następująco (korzystając z techniki dziel i zwyciężaj):

1. posortuj lewą połowę tablicy (rekurencyjnie),
2. posortuj prawą połowę tablicy (rekurencyjnie),
3. scal posortowanie półówki w jedną, posortowaną tablicę

Proszę zaimplementować funkcję `mergesort(T)`, która sortuje tablicę T przy pomocy powyższego algorytmu (oczywiście należy uzupełnić szczegóły, w tym warunki brzegowe rekurencji).