**Porządkowanie przez scalanie**

**Dane: Liczba naturalna: n>0 (liczba elementów listy T)**

**n-elementowa lista jednowymiarowa zawierająca liczby rzeczywiste: T[0…n -1].**

**[4, 5, 2, 1, 9, 8, 7, 0, 3]**

**Wyniki: Posortowana niemalejąco n-elementowa lista jednowymiarowa zawierająca liczby rzeczywiste: T[0…n – 1]**

**Lista kroków**

**Algorytm rekurencyjny (T, n, lewy, prawy)**

**Krok 1.**

**Jeśli lewy<prawy, przypisz srodek=lewy + prawy / 2 i przejdź do kroku 2., w przeciwnym wypadku zakończ algorytm**

**Krok 2.**

**Uruchom ten algorytm z parametrami (T, lewy, srodek).**

**Krok 3.**

**Uruchom ten algorytm z parametrami (T, srodek +1, prawy).**

**Krok 4.**

**Scal ciągi T[lewy, srodek] i T[srodek +1, prawy] w jeden ciąg T[lewy, prawy].**

**# sortowanie przez scalanie - rekurencja**