

Lista numer 5: algorytmy sortowania (12 pkt.)

Zadanie 0

- Zaimplementować klasę Student, która posiada następujące pola: numer indeksu, nazwisko, imię, miesięczna kwota stypendium przyznanego studentowi.
- Zaimplementować w klasie student dwie metody:
 - `public int compareTo(Object drugi),`
 - `public boolean equals(Object drugi),`wykorzystujące do porównania studenta numery indeksów.

Zadanie 1 (3 pkt.)

- Zaimplementować następujące komparatory:
 - komparator naturalny porównujący dwa obiekty typu student na podstawie numeru indeksu,
 - komparator złożony wykorzystujący listę jednokierunkową do przechowywania zestawu komparatorów,
 - komparator porównujący dwa obiekty typu student na podstawie imienia,
 - komparator porównujący dwa obiekty typu student na podstawie nazwiska,
 - komparator porównujący dwa obiekty typu student na podstawie przyznanego stypendium,
 - komparator odwrotny.

Zadanie 2 (3 pkt)

- Zaimplementować dwa z trzech omówionych na wykładzie prostych algorytmów sortowania: sortowanie bąbelkowe, przez wybór, przez wstawianie, wykorzystując dowolną strukturę danych (np. listę, tablicę) do przechowywania informacji o studentach.
- Podczas sortowania zliczać ilość porównań (wywołań komparatora) oraz ilość operacji przestawień.
- Stworzyć listę studentów ułożonych w sposób przypadkowy. Zaprezentować działanie wybranego algorytmu sortowania, który posortuje studentów wg:
 - numeru indeksu rosnąco,
 - stypendium malejąco,
 - nazwiska, a jeśli są identyczne, to imienia, malejąco.

Zadanie 3 (3 pkt.)

- Zaimplementować dwa z trzech omówionych na wykładzie złożonych algorytmów sortowania: sortowanie Shella, szybkie, przez scalanie, wykorzystując dowolną strukturę danych (np. listę, tablicę) do przechowywania informacji o studentach.
- Podczas sortowania zliczać ilość porównań (wywołań komparatora) oraz ilość operacji przestawień.

Zadanie 4 (3 pkt)

- Dla każdego z zaimplementowanych algorytmów sortowania wykonać sortowanie rosnące wg numeru indeksu ciągu obiektów typu student o rozmiarze: 100, 250, 500, 750, 1000, zakładając, że ciąg ten na początku jest:
 - losowy,

- posortowany malejąco (przypadek pesymistyczny),
 - rosnąco (przypadek optymistyczny).
- Dla każdego algorytmu należy zliczyć ilość porównań p , ilość zamian z , w zależności od ilości elementów n oraz przypadku. Przykładowo, dla klasy Student: numer indeksu należy wylosować, imię i nazwisko to może być pojedyncza litera alfabetu wybrana losowo, stypendium może być także losowane.
- Przygotować sprawozdanie zawierające 3 wykresy przedstawiające $p(n)$ oraz 3 wykresy przedstawiające $z(n)$ dla badanych algorytmów, dla 3 przypadków:
 - losowego,
 - pesymistycznego,
 - optymistycznego.

Skomentować krótko w sprawozdaniu złożoność poszczególnych algorytmów.