

## 3.7 Różne sposoby sortowania

Co robimy na lekcji:

### Różne PRZYKŁADY SORTOWANIA

- **Sortowanie bąbelkowe (bubblesort)**

To jeden z najprostszych algorytmów porządkujących.

Stosując ten algorytm podczas porządkowania np. tablicy zawierającej przypadkowo rozmieszczone liczby, które chcemy uporządkować rosnąco (czyli od najmniejszej liczby do największej) – “przechodzimy” od początku tablicy do jej końca i porównujemy dwie sąsiednie liczby. Jeżeli pierwsza liczba jest większa od drugiej, to zamieniamy je miejscami.

Po pierwszym “przejściu” wykonujemy kolejne itd.

Sortowanie kończy się, gdy podczas kolejnego przejścia nie dokonujemy żadnej zmiany.

- Jeżeli obejrzysz poniższy film, to zobaczysz występ węgierskiego zespołu ludowego, którego członkowie pokazują tańcem, jak realizowany jest algorytm sortowania bąbelkowego

[Bubble-sort with Hungarian \("Csángó"\) folk dance](#)

6 5 3 1 8 7 2 4

---

Algorytm nosi nazwę bąbelkowego, ponieważ przypomina przemieszczanie się bąbelków w wodzie, ku górze.

- **Sortowanie przez wstawianie (Insert Sort, Insertion Sort)**

Metoda ta polega na układaniu kolejnych elementów poprzez wstawianie ich we właściwe miejsce, co przypomina układanie kart do gry w dłoni, kiedy każdą kartę umieszczamy w odpowiednim miejscu, tzn. po młodszej, ale przed

---

starszą.

Stosując ten algorytm w celu uporządkowania jakiegoś zbioru elementów (np. przypadkowych liczb) – dzielimy zbiór na dwie części: część uporządkowaną (początkowo pustą) i część nieuporządkowaną (zawierającą początkowo wszystkie elementy zbioru, który chcemy uporządkować).

Zaczynamy od przeniesienia dowolnie wybranego elementu z części nieuporządkowanej do części posortowanej.

Potem kolejny element przenosimy z części nieuporządkowanej do części uporządkowanej i porównujemy go ze znajdującym się tam elementem – oba ustawiamy we właściwej kolejności.

Następnie kolejny element przenosimy z części nieuporządkowanej do części uporządkowanej i porównujemy go z kolejnymi elementami zbioru uporządkowanego, dopóki nie trafi we właściwe miejsce.

Podobnie postępujemy z kolejnymi elementami.

Algorytm zostaje zakończony, gdy wszystkie elementy znajdują się w zbiorze uporządkowanym.

Taneczną ilustrację działania tego algorytmu możesz zobaczyć na poniższym filmie

[Insert-sort with Romanian folk dance](#)

- **Sortowanie przez scalanie (merge sort)**

Stosując ten algorytm w celu uporządkowania jakiegoś zbioru elementów (np. przypadkowych liczb), zaczynamy od podzielenia danego zbioru na mniejsze zbiory (najpierw dzielimy zbiór na dwie części, potem każdą z tych części na kolejne dwie itd.), aż do uzyskania zbiorów jednoelementowych (które są – oczywiście – posortowane).

Następnie kolejno łączymy zbiory jednoelementowe w dwuelementowe – ustawiając ich elementy we właściwej kolejności; a potem dwuelementowe w czteroelementowe – znowu ustawiając ich elementy we właściwej kolejności... itd.

Algorytm zostaje zakończony, gdy wszystkie mniejsze zbiory zostaną połączone, a ich elementy – posortowane

Poniżej – taneczna wizualizacja działania tego algorytmu

[Merge-sort with Transylvanian-saxon \(German\) folk dance](#)

- Gotowy projekt w Scratch – SORTOWANIE do naszego tematu z podręcznika - zobacz

[lekcja3.7\\_program on Scratch \(mit.edu\)](#)

Porządkowanie przez zliczanie

- [Zapoznaj się z tematem z podręcznika 3.6](#)
- [Temat 3.7 sortowanie](#)
- Obejrzyj film – przynajmniej 3 pierwsze minuty a dowiesz się na czym polega algorytm sortowania przez zliczanie
- [Informatyka SP4 8 kl 7 zm Lekcja 3 6](#)
- Jeżeli jesteś ciekawy jak to zrobić w scratch - obejrzyj cały film.

Jak to wygląda w scratch – zobacz

[porządkowanie przez zliczanie on Scratch \(mit.edu\)](#)

ZADANIE /podręcznik

Zwiększ liczbę losowanych cyfr dwukrotnie. Musisz je ustawić tak, aby wszystkie były widoczne, np. w dwóch wierszach jeden pod drugim.

Zobacz – jak to powinno być zrobione

[lekcja3.6\\_zadanie1\\_program on Scratch \(mit.edu\)](#)

Z dzisiejszej lekcji nic nie przesyłacie do sprawdzenia

Przeczytaj podsumowanie działu s. 108

**- WYKONAJ KRÓTKI TEST - SPRAWDZENIE WIADOMOŚCI O ALGORYTMACH I SCRATCHU /OBOWIAZKOWY**

**Test jest na 15- 20 minut - można go wykonać tylko raz - jeżeli jesteś gotowa/gotowy/ wykonaj**

**Jeeli masz mało czasu niezaczynaj - zrób na następnej lekcji**

**<https://forms.office.com/r/py6kVP7L7W>**

**termin do 10 maja 2022r.**