

### 3. Algorytmy porządkowania

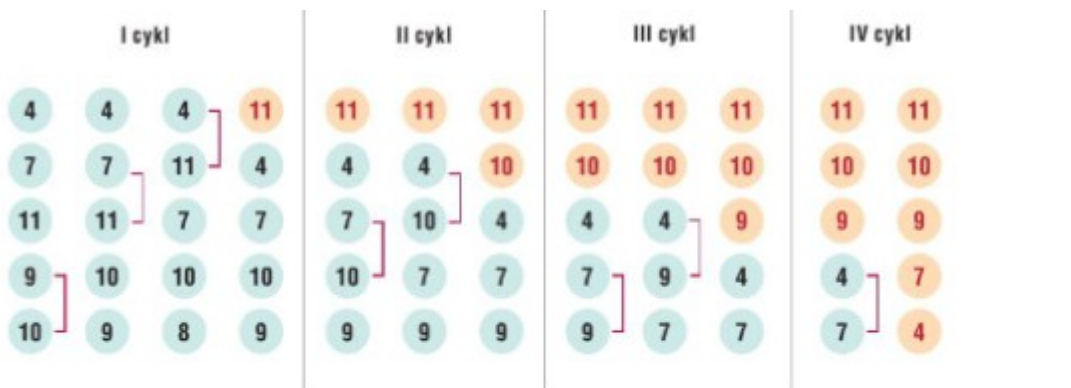
{ Chcemy uporządkować malejąco wprowadzone z klawiatury liczby.  
W jaki sposób napisać program porządkujący elementy ciągu liczb? }

Elementy ciągu liczb możemy uporządkować, stosując wybrany algorytm sortowania (porządkowania), np. **metodę bąbelkową** lub **przez wstawianie**.

#### 3.1. Porządkowanie metodą bąbelkową

W algorytmie sortowania metodą bąbelkową w celu uporządkowania ciągu liczb – w tym przypadku malejąco – porównujemy sąsiednie liczby, rozpoczynając od porównania ostatniej i przedostatniej. Jeśli liczba ostatnia jest większa od przedostatniej, zamieniamy je miejscami.

Następnie liczbę, która stała się przedostatnią, porównujemy z trzecią od końca i przestawiamy, jeśli jest większa itd.



oznacza zamianę elementów miejscami

Rys. 2. Przykład zastosowania algorytmu sortowania bąbelkowego

Na rysunku 2. pokazano przykład sortowania bąbelkowego pięciu liczb – sortowanie przebiega od dołu ku górze. Po pierwszym cyklu ciąg nie jest jeszcze uporządkowany. Należy wykonywać kolejne cykle, aż ciąg zostanie uporządkowany i nie trzeba już będzie przestawić żadnego elementu.



**W algorytmie sortowania bąbelkowego** porównujemy parami kolejne elementy zbioru i przestawiamy je, jeśli są ustawione w niewłaściwej kolejności.

Zbiór jest przeglądany w tym samym kierunku tak długo, jak długo występuje w nim para elementów ustawionych w niewłaściwym porządku.

Nietrudno zauważyć, że już w pierwszym cyklu sortowania liczba największa (w tym przypadku liczba 11) zostaje ustawiona na właściwym miejscu. Aby ustawić we właściwej kolejności pozostałe liczby, należy powtarzać cykl sortowania, dopóki nie zostaną wykonane wszystkie przestawienia. Rysunek 2. jest poglądową ilustracją pokazującą działanie algorytmu sortowania bąbelkowego. Elementy o większych wartościach są wynoszone wyżej, co przypomina bąbelki powietrza (większe szybciej przemieszczają się ku górze) – stąd też wzięła się nazwa algorytmu.

Oczywiście tą metodą możemy uporządkować zbiór, zaczynając od góry, czyli od porównania elementu pierwszego z drugim, drugiego z trzecim itd.



#### Ćwiczenie 5. Stosujemy sortowanie bąbelkowe

Uporządkuj metodą bąbelkową malejąco (od największej do najmniejszej) liczby 34, 87, 45, 12, 11, 0, -56, -28, -11, 72. Podaj liczbę porównań, wykonanych w celu uporządkowania tego ciągu.

##### Wskazówki:

- Przygotuj kartki z zapisanymi przykładowymi liczbami.
- Ułóż kartki na stole lub na podłodze i przestawiaj parami. Zacznij od ostatniej.

## 3.2. Porządkowanie przez wstawianie

Metodę sortowania (porządkowania) **przez wstawianie** (inaczej **przez umieszczanie**) stosujemy często w życiu codziennym – na przykład gdy odkładamy na półkę książkę w bibliotece, znajdujemy odpowiednie miejsce, patrząc na nazwiska autorów książek już stojących na półce.

W metodzie sortowania przez wstawianie porządkowany ciąg dzielimy na dwie części (rys. 3.):

- uporządkowaną (na początku działania algorytmu nie zawiera ona żadnych elementów),
- nieuporządkowaną (na początku zawiera wszystkie elementy ciągu).

Z części nieuporządkowanej pobieramy kolejne elementy i umieszczamy je w odpowiednich miejscach w części uporządkowanej.

Aby znaleźć miejsce umieszczenia nowego elementu w ciągu uporządkowanym, zastosujemy metodę **liniową**, w której porównuje się wstawiany element z kolejnymi elementami ciągu uporządkowanego.



#### Ćwiczenie 7. Analizujemy algorytm sortowania metodą przez wstawianie

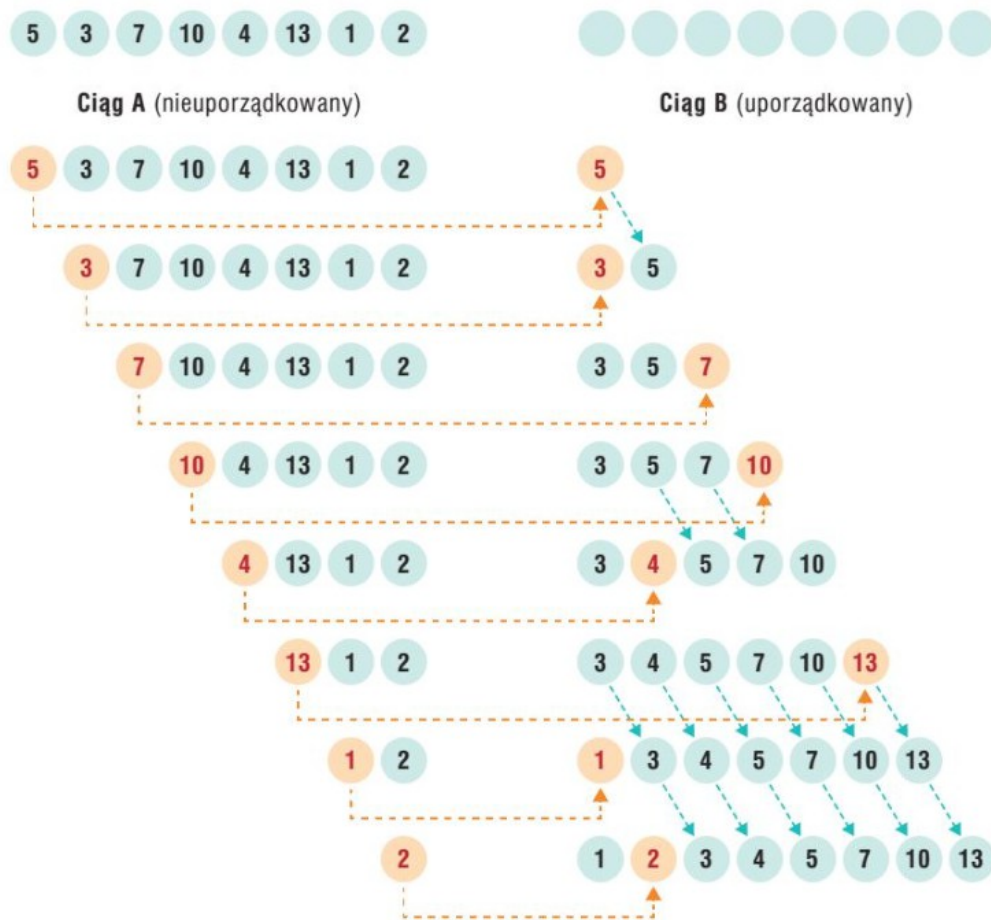
Przeanalizuj algorytm porządkowania przez wstawianie, wykonując go dla ciągu (3, 4, 10, 7, 1, 5, 2, 8).

**Wskazówka:** Przygotuj niezbędne pomoce dydaktyczne, np. kartoniki z zapisanymi liczbami.



#### Ćwiczenie 8. Sprawdzamy liczbę porównań elementów

Podaj liczbę porównań, wykonanych w celu uporządkowania ciągu z ćwiczenia 7.



Rys. 3. Przykład działania algorytmu sortowania przez wstawianie

Temat do samodzielnej realizacji, bez łączenia za pośrednictwem Teams.

E.O.