

## Lekcja 48 (kl. II. PR)

**Temat:** Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa.

### Cele lekcji:

- poznanie istoty sortowania bąbelkowego,
- ułożenie algorytmu realizującego sortowanie bąbelkowe

### Uczeń:

- omawia działanie metody bąbelkowej na przykładzie;
- zespołowo pracuje nad projektem algorytmu opisującego metodę bąbelkową;
- przeprowadza weryfikację poprawności działania algorytmu.

Podręcznik str. 287 – Informatyka. 2. PR, Operon

### **Przebieg lekcji:**

1. Zapoznanie się z celami lekcji.
2. Opis metody, czyli na czym polega sortowanie bąbelkowe – film - <https://www.youtube.com/watch?v=4s44rXRdmhQ>
3. Sortowanie bąbelkowe w praktyce, czyli układamy algorytm – podręcznik, rys. 47.1
4. Ćwiczenia praktyczne – stosowanie sortowania bąbelkowego w programowaniu w C++.

### **Zadania do wykonania - str. 292**

---

#### Sortowanie bąbelkowe

[powrót](#)

**Sortowanie bąbelkowe** jest jednym z najprostszych w implementacji algorytmów porządkujących dane. Złożoność tego sposobu sortowania jest rzędu

$$O(n^2)$$

. Oznacza to, że sortowanie bąbelkowe nie poradzi sobie z porządkowaniem większych zbiorów.

Nazwa wzięła się stąd, że dane podczas sortowania - tak jak bąbelki w napoju gazowanym - przemieszczają się ku prawej stronie i układają się w odpowiednim szyku.

Algorytm działa następująco:

w każdym przejściu pętli wewnętrznej porównywane są ze sobą dwie kolejne wartości i w razie potrzeby są zamieniane miejscami. W jednym cyklu pętli wewnętrznej, największa liczba (tak jak bąbelki w napoju gazowanym) w zbiorze będzie się przemieszczała na ostatnią pozycję. W ten sposób otrzymujemy podzbiór częściowo już posortowany. Czynności te powtarzamy dla zbioru pominiętego o elementy już poukładane. Prześledźmy przykład:

posortujemy tą metodą następujące dane

3 2 4 3 1 2 0

Pętla wewnętrzna porówna sąsiadujące ze sobą elementy. Dla

$n$

elementów zostanie wykonanych

$n - 1$

porównać:

$\overset{23}{\frown}$   
3 2 4 3 1 2 0

$\overset{34}{\frown}$   
2 3 4 3 1 2 0

$\overset{34}{\frown}$   
2 3 4 3 1 2 0

$\overset{14}{\frown}$   
2 3 3 4 1 2 0

$\overset{24}{\frown}$   
2 3 3 1 4 2 0

$\overset{04}{\frown}$   
2 3 3 1 2 4 0

sytuacja po pierwszym cyklu pętli wewnętrznej:

2 3 3 1 2 0 4

Największa liczba (w tym przypadku

4

) przesunęła się na ostatnią pozycję. Został nam teraz do posortowania zbiór elementów pominięty o tą liczbę. Powtarzamy teraz czynności dla

$n - 1$

elementów. W ten sposób liczba

3

wskoczy na przedostatnią pozycję. Powtarzamy te kroki, aż otrzymamy zbiór posortowany.

```
#include<iostream>
using namespace std;
```

```
void sortowanie_babelkowe(int tab[],int n)
{
    for(int i=0;i<n;i++)
        for(int j=1;j<n-i;j++) //pętla wewnętrzna
            if(tab[j-1]>tab[j])
                //zamiana miejscami
                swap(tab[j-1], tab[j]);
}
```

```
int main()
{
    int *tab, n;

    cout<<"Ile liczb będziesz chciał posortować? ";
    cin>>n;

    tab = new int [n]; //przydzielenie pamięci na elementy tablicy
    //wczytanie liczb
    for(int i=0;i<n;i++)
        cin>>tab[i];

    sortowanie_babelkowe(tab,n);

    //wypisanie posortowanych elementów
    for(int i=0;i<n;i++)
        cout<<tab[i]<<" ";

    return 0;
}
```