



---

## ĆWICZENIA INFORMATYKA I

### Lista nr 4. Schematy blokowe. Tablice. Sortowanie.

---

1. Narysuj schemat blokowy algorytmu, który dla tablicy liczb całkowitych o stałym rozmiarze:
  - a) znajduje wartość największą i najmniejszą,
  - b) oblicza średnią arytmetyczną,
  - c) oblicza standardowe odchylenie
2. Narysuj schemat blokowy algorytmu, który "przesuwa" elementy tablicy o jedno miejsce w prawo albo w lewo (wybiera użytkownik) wolne miejsca uzupełniając zerami, np. tablica (5,4,3,2,1) przesunięta w prawo to (0,5,4,3,2).
3. Narysuj schemat blokowy algorytmu, który oblicza pochodną wielomianu. Wynik działania, czyli pochodną, algorytm ma zapisywać do tablicy.
4. **Sortowanie bąbelkowe.** Narysuj schemat blokowy algorytmu, który posortuje tablicę liczb losowych rosnąco według podanego algorytmu: sortowanie bąbelkowe polega na porównywaniu dwóch kolejnych elementów i zamianie ich kolejności, jeżeli zaburza ona porządek, w jakim się sortuje tablicę. Sortowanie kończy się, gdy podczas kolejnego przejścia (porównania każdego elementu tablicy z następnym) nie dokonano żadnej zmiany.  
Przykład działania dla tablicy (4,2,1,7,5):  
1 przejście tablicy:  
(4,2,1,7,5) → zamiana miejscami 2 i 4 → (2,4,1,7,5) → zamiana miejscami 4 i 1 → (2,1,4,7,5) → (2,1,4,7,5) → zamiana miejscami 7 i 5 → (2,1,4,5,7)  
2 przejście tablicy:  
(2,1,4,5,7) → zamiana miejscami 2 i 1 → (1,2,4,5,7) → (1,2,4,5,7) → (1,2,4,5,7)  
3 przejście tablicy:  
(1,2,4,5,7) → (1,2,4,5,7) → (1,2,4,5,7) → (1,2,4,5,7)  
Koniec działania algorytmu, ponieważ tablica jest posortowana.
5. **Sortowanie przez wybieranie.** Narysuj schemat blokowy algorytmu, który posortuje tablicę liczb losowych rosnąco według podanego algorytmu: jeśli sortujemy tablicę n-elementową rosnąco, to na i-tym miejscu ma znaleźć się najmniejszy element znaleziony wśród elementów od i do n tablicy, co oznacza, że porównujemy i-ty element z minimum znalezionym wśród elementów od i+1 do n i zamieniamy miejscami te wartości, jeśli i-ty element jest większy niż znalezione minimum.  
Przykład działania dla tablicy (4,5,3,7,2) (n=5, algorytm będzie miał 4 kroki):  
1 krok:  
(4,5,3,7,2) → zamieniamy wartość 4 z wartością 2 → (2,5,3,7,4)  
2 krok:  
(2,5,3,7,4) → zamieniamy wartość 5 z wartością 3 → (2,3,5,7,4)  
3 krok:  
(2,3,5,7,4) → zamieniamy wartość 5 z wartością 4 → (2,3,4,7,5)  
4 krok:

$(2,3,4,7,5) \rightarrow$  zamieniamy wartość 7 z wartością 5  $\rightarrow (2,3,4,5,7)$ .

6. **Sortowanie przez wstawianie.** Narysuj schemat blokowy algorytmu, który posortuje tablicę liczb losowych rosnąco według podanego algorytmu: wybieramy po kolei elementy (zaczynamy od drugiego), porównujemy go z kolejnymi elementami zbioru już posortowanego (w pierwszym kroku zbiór posortowany zawiera tylko jeden element), póki nie napotkamy elementu równego lub elementu większego (jeśli chcemy otrzymać ciąg rosnący) lub nie znajdziemy się na początku/końcu zbioru uporządkowanego. Wyciągnięty element wstawiamy w miejsce gdzie skończyliśmy porównywanie.

Przykład działania dla tablicy  $(4,10,3,7,1)$  (5 elementów, więc 4 kroki są potrzebne):

1 krok:

$(4,10,3,7,1) \rightarrow$  wartość 10 jest na właściwym miejscu, bo jest większa niż 4

2 krok:

$(4,10,3,7,1) \rightarrow$  wartość 3 trzeba wstawić w odpowiednie miejsce, porównujemy tą wartość z elementami ją poprzedzającymi, ma być na pierwszym miejscu, więc elementy drugi i trzeci przesuwamy w prawo i na miejsce pierwsze wstawiamy wartość 3  $\rightarrow (3,4,10,7,1)$

3 krok:

$(3,4,10,7,1) \rightarrow$  wartość 7 trzeba wstawić w odpowiednie miejsce, porównujemy tą wartość z elementami ją poprzedzającymi, ma być na trzecim miejscu, więc element czwarty przesuwamy w prawo i na miejsce trzecie wstawiamy wartość 7  $\rightarrow (3,4,7,10,1)$

4 krok:

$(3,4,7,10,1) \rightarrow$  wartość 1 trzeba wstawić w odpowiednie miejsce, porównujemy tą wartość z elementami ją poprzedzającymi, ma być na pierwszym miejscu, więc pozostałe elementy przesuwamy w prawo i na miejsce pierwsze wstawiamy wartość 1  $\rightarrow (1,3,4,7,10)$ .

7. Narysuj schemat blokowy algorytmu, który realizuje wyszukiwanie binarne zadanej wartości w posortowanej tablicy liczb całkowitych i jeśli się ona w tablicy znajduje, podaje numer elementu o tej wartości. Wyszukiwanie binarne polega na dzieleniu tablicy na coraz mniejsze przedziały do momentu, gdy szukany element zostanie znaleziony, bądź przedział osiągnie długość zero, co oznacza brak szukanego elementu. W każdym kroku wyznaczany jest środek przedziału i sprawdzane jest, czy element zapisany na środku przedziału jest poszukiwanym elementem (wówczas algorytm kończy działanie), czy jest on niego mniejszy (albo większy): wówczas przedział jest odpowiednio zawężany.