

5.1. Złożoność czasowa

Złożoność czasową możemy wyrazić również w jednostkach czasu. Ale nie jest to wygodne – ze względu na różne procesory, w które mogą być wyposażone komputery, czas wykonywania podobnych operacji może się różnić.

Złożoność czasowa algorytmu zależy od liczby operacji niezbędnych dla ukończenia algorytmu. W przypadku algorytmów uniwersalnych oczywiste jest, że złożoność czasowa algorytmu zależy będzie od wielkości zbioru danych, na jakich pracuje algorytm. A zatem złożoność czasowa jest pewnego rodzaju funkcją wielkości zbioru danych, stąd często w literaturze oznacza się ją symbolem $O(f(n))$, gdzie $f(n)$ – pewna funkcja, np. $O(n)$, $O(n^2)$, $O(n!)$. Szacowanie złożoności czasowej jest trudnym zadaniem, postaramy się jednak przybliżyć je na przykładach.

W poznanych algorytmach, np. wyboru minimum, sortowania bąbelkowego, sortowania przez wybór, podstawową operacją jest porównywanie dwóch elementów. Jako kryterium oceny złożoności czasowej tych algorytmów przyjmujemy liczbę wykonywanych porównań.



Przykład 9. Wyszukiwanie liczby w tablicy liczb

Zadanie: Wyszukaj liczbę x w tablicy n liczb.

Dane: Tablica liczb $a[1..n]$, gdzie n – liczba elementów tablicy, x – szukana liczba.

Wynik: Numer indeksu pierwszego elementu tablicy $a[1..n]$, równego szukanej liczbie x , lub -1 , jeżeli liczba x nie zostanie znaleziona.

Uwaga: W przykładzie 9. dla uproszczenia przyjęto indeksowanie elementów tablicy od 1 (tak jest w języku Pascal). Przypominamy, że w języku C++ indeksowanie elementów tablicy zaczyna się od zera.

Lista kroków:

1. Zaczynaj algorytm.
2. Zmiennej i przypisz wartość 1: $i := 1$.
3. Jeśli $i > n$, przejdź do kroku 7, zwracając -1 (element nie został odnaleziony).
4. Jeśli $a[i] = x$, przejdź do kroku 7, zwracając i .
5. Powiększ wartość zmiennej i o 1: $i := i + 1$.
6. Przejdź do kroku 3.
7. Zakończ algorytm.



Ćwiczenie 7.

Wykonaj algorytm podany w przykładzie 9, wyszukując w tablicy liczb $[12, 11, 8, 6, 8, 4, 10, 5, 2, 3]$ liczby: 7, 8, 3, 1. Ile operacji porównania musiałeś wykonać każdorazowo? Ile ich wykonałeś średnio?

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

Podstawową operacją dla algorytmu podanego w przykładzie 9. jest operacja porównania. W skrajnych przypadkach liczba operacji porównań L_n może być równa 1 ($L_n = 1$), gdy pierwszy element tablicy $a[1]$ jest równy x , lub może być równa n ($L_n = n$), gdy element x nie występuje w tablicy (aby to ustalić, należy porównać x ze wszystkimi elementami tablicy, a więc wykonać n porównań). Przy rozpatrywaniu złożoności czasowej algorytmu istotny jest rząd wielkości liczby wykonywanych operacji, a nie dokładna wartość tej liczby, która jest zależna od konkretnego zbioru danych. Dla algorytmu wyszukiwania rząd ten wynosi n , a zatem złożoność czasowa algorytmu to $O(n)$ (mówimy, że złożoność czasowa algorytmu jest rzędu n). Dokładne wyjaśnienie tego zagadnienia można znaleźć w literaturze fachowej.