



# **Politechnika Krakowska** **im. Tadeusza Kościuszki**

## **Matematyka dyskretna**

Dokumentacja zadania – zad nr 2)

Gabriela Rączka

Nr albumu: 147085

gr nr 02

Zad 1) Opisz na czym polega algorytm przeszukiwania binarnego (ang. *binary search*), zwracając szczególną uwagę na użytą w nim rekurencję. Utwórz w dowolnym języku programowania program wykorzystujący wspomniany algorytm.

Algorytm *binary search* to algorytm wyszukiwania wartości w posortowanej tablicy. Początkowo dzieląc tablicę na pół, algorytm określa, czy pożądana wartość leży w lewej, czy w prawej połowie. Proces jest kontynuowany rekurencyjnie, dopóki zapytanie nie zostanie rozwiązane lub udowodnione, że nie istnieje w tablicy.

Utworzony w języku C++ program pobiera od użytkownika rozmiar tablicy oraz wartości dla tablicy, sortuje tablicę przy użyciu funkcji *sort* z biblioteki *algorithm*, a następnie pobiera wartość, którą użytkownik chce wyszukać w tablicy. Wyszukiwanie jest realizowane za pomocą funkcji *binarySearch*, która implementuje algorytm wyszukiwania binarnego. W przypadku znalezienia elementu w tablicy, program wyświetla jego indeks, w przeciwnym wypadku wyświetla komunikat o nieznalezieniu elementu.

Konkretnie, funkcja *binarySearch* przyjmuje jako argumenty:

- **arr** - wskaźnik na tablicę do przeszukania,
- **left** - indeks lewej granicy przeszukiwania,
- **right** - indeks prawej granicy przeszukiwania,
- **target** - szukany element wprowadzony przez użytkownika.

W pierwszej części funkcja sprawdza, czy granice nie zostały przekroczone (czyli czy są poprawne). Jeśli nie, zwraca -1, co oznacza, że szukany element nie został znaleziony w tablicy. W przeciwnym wypadku wylicza indeks elementu środkowego tablicy i porównuje go z szukany elementem. Jeśli element środkowy jest równy szukanemu, to zwraca jego indeks. W przeciwnym wypadku jeśli element środkowy jest większy od szukanego, funkcja rekurencyjnie wywołuje się na lewej części tablicy (od lewej granicy do elementu środkowego minus jeden). W przeciwnym wypadku wywołuje się na prawej części tablicy (od elementu środkowego plus jeden do prawej granicy). Jeśli funkcja nie znajdzie szukanego elementu w tablicy, zwraca -1.

W funkcji *main*, do tablica *arr* zostają przypisane elementy wprowadzone przez użytkownika, które zostają posortowane w tablicy przy pomocy funkcji *sort*. Wartość zmiennej *size* określa liczbę elementów w tablicy, którą podaje użytkownik (tablica została zainicjalizowana tak, że zmienna *size* musi być mniejsza od 100). Następnie funkcja *binarySearch* jest wywoływana na całej tablicy i szukanym elemencie *target*. Wynik jest przypisywany do zmiennej *result*.

Zad 2) Opisz na czym polega algorytm sortowania przez scalanie (ang. *merge sort*), zwracając szczególną uwagę na użytą w nim rekurencję. Utwórz w dowolnym języku programowania program wykorzystujący wspomniany algorytm.

Algorytm *merge sort* to algorytm sortowania danych, który opiera się na strategii "*dziel i zwyciężaj*". Ta metodologia wymaga podzielenia listy na pół, niezależnego sortowania każdej z dwóch części, a następnie połączenia ich w jedną posortowaną listę. Proces powtarza się rekurencyjnie, aż poszczególne elementy zostaną osiągnięte i uznane za posortowane automatycznie.

Program został napisany w języku C++. Na początku program prosi użytkownika o podanie liczby elementów do posortowania oraz pobiera elementy z wejścia. Następnie wywołuje funkcję *MergeSort* zaczynając sortowanie od całego zakresu elementów od 0 do n-1.

Funkcja *MergeSort* to rekurencyjna funkcja, która dzieli tablicę na pół, sortuje obie połowy z osobna wywołując się rekurencyjnie dla każdej z nich, a następnie łączy je w jedną posortowaną tablicę za pomocą funkcji *Merge*.

Funkcja *Merge* działa na posortowanych podtablicach *array[low..mid]* i *array[mid+1..high]*. Tworzy tymczasową tablicę *temp*, porównuje wartości elementów z dwóch podtablic, i umieszcza mniejszy z nich w tablicy *temp*. Następnie składa dwie podtablice w jedną posortowaną tablicę *array[low..high]*.

Na końcu program wyświetla posortowane elementy i zwalnia zaalokowaną pamięć na tablicę.