

## **ZADANIE 2 – ZŁOŻONE STRUKTURY DANYCH**

prowadząca: prof. dr hab. inż. Małgorzata Sterna

1. Dla minimum 10 wartości  $n$  należy wygenerować tablicę  $A$  (statyczną implementację listy) zawierającą  $n$  niepowtarzających się liczb całkowitych dodatnich.
2. Wykonać kopię tablicy  $A$  – tablicę  $B$  – i posortować ją w porządku rosnącym korzystając z metody sortowania szybkiego. Zmierzyć czas tworzenia tablicy –  $C_B$  (tworzenia kopii i sortowania). Zmierzyć czas wyszukiwania kolejnych elementów tablicy  $B$  w tablicy  $A$  (wyszukiwanie w statycznej implementacji listy) –  $S_A$  oraz czas wyszukiwania kolejnych elementów tablicy  $A$  w tablicy  $B$  na zasadzie dzielenia połówkowego (wyszukiwanie binarne w tablicy posortowanej) –  $S_B$ .
3. Zbudować drzewo BST – drzewo  $TA$  – wprowadzając elementy w kolejności podanej w tablicy  $A$ , mierząc czas jego tworzenia  $C_{TA}$ . Zmierzyć jego wysokość  $h_{TA}$ . Zmierzyć czas wyszukiwania kolejnych elementów tablicy  $A$  w drzewie  $TA$  –  $S_{TA}$ .
4. Zbudować drzewo BST – drzewo  $TB$  – wprowadzając elementy w kolejności wynikającej z dzielenia połówkowego tablicy  $B$ , mierząc czas jego tworzenia  $C_{TB}$ . (Elementy należy umieścić w tablicy pomocniczej aby czas dzielenia połówkowego nie był wliczony w  $C_{TB}$ ). Zmierzyć jego wysokość  $h_{TB}$ . Zmierzyć czas wyszukiwania kolejnych elementów tablicy  $A$  w drzewie  $TB$  –  $S_{TB}$ .

Przedstawić w tabeli oraz na wspólnym wykresie zależność czasu tworzenia struktur od liczby elementów, czyli  $t=C_B(n)$ ,  $t=C_{TA}(n)$ ,  $t=C_{TB}(n)$ .

Przedstawić w tabeli oraz na wspólnym wykresie zależność czasu wyszukiwania od liczby elementów, czyli  $t=S_A(n)$ ,  $t=S_B(n)$ ,  $t=S_{TA}(n)$ ,  $t=S_{TB}(n)$ .

Przedstawić w tabeli oraz na wspólnym wykresie zależność wysokości drzew od liczby elementów, czyli  $h = h_{TA}(n)$ ,  $h = h_{TB}(n)$ .

Wyciągnąć wnioski z eksperymentu, m.in.:

- jaka jest wysokość drzew  $TA$  i  $TB$  funkcji  $n$ ? jaka byłaby wysokość drzewa, gdyby wprowadzano elementy wg tablicy  $B$ ? które drzewo jest najkorzystniejsze, a które najmniej korzystne? dlaczego?
- jaki jest powód różnicy wysokości między drzewami  $TA$  i  $TB$ ?
- czy wysokość drzewa poszukiwań binarnych jest istotna? jak wpływa na nią uporządkowanie danych wejściowych? jaka jest złożoność procedury tworzenia drzewa i wyszukiwania wszystkich elementów? w jakim celu i kiedy stosuje się wyważanie drzewa BST?
- porównaj wyszukiwanie w tablicy  $B$  z wyszukiwaniem w tablicy  $A$  (statycznej implementacji listy)
- porównaj wyszukiwanie połówkowe w tablicy  $B$  z wyszukiwaniem w drzewie  $TB$
- porównaj wyszukiwanie połówkowe w tablicy  $B$  z wyszukiwaniem w drzewie  $TA$
- porównaj zajętość pamięciową poszczególnych struktur, podaj jaki jest (fizyczny) rozmiar pojedynczego elementu tablicy i drzewa BST
- jakie są wady i zalety tablicy, listy jednokierunkowej i drzewa BST wykorzystanych do wyszukiwania informacji (można uwzględnić np. zajętość pamięciową, efektywność czasową, łatwość implementacji, dopisywanie, usuwanie elementów itp.)?