

## Ćwiczenia 15

*Zadanie 2, kolokwium 2, 2021/22*

W tym zadaniu rozważamy dynamiczny, różnowartościowy ciąg liczb całkowitych  $C$ . Na ciągu  $C$  wykonujemy operacje:

- $\text{Ini}(C):: C := []$  // inicjacja ciągu pustego — tylko raz, na początku wykonywania wszystkich operacji
- $\text{Insert}(C, x, i)::$  wstaw element  $x$  na pozycję  $i$  w ciągu  $C$ ,  $1 \leq i \leq |C| + 1$ , // możesz przyjąć, że  $x$  nie ma w ciągu  $C$
- $\text{Delete}(C, i)::$  usuń  $i$ -ty element z ciągu  $C$ ,  $1 \leq i \leq |C|$ ,
- $\text{Sorted}(C, i, j)::$  sprawdź, czy podciąg  $C_i, C_{i+1}, \dots, C_j$  jest uporządkowany rosnąco,  $1 \leq i \leq j \leq |C|$ ,
- $\text{BSComp}(C, i, j)::$  sprawdź, czy elementy  $C_i$  oraz  $C_j$  ( $1 \leq i \leq j \leq |C|$ ) byłyby porównywane ze sobą w pierwszym przebiegu sortowania tablicy  $a[1..|C|] = [C_1, C_2, \dots, C_{|C|}]$  algorytmem BubbleSort, który wykonuje pętlę **for**  $k = 1, 2, \dots, |C| - 1$  **do** **if**  $a[k] > a[k + 1]$  **then**  $a[k] := a[k + 1]$ .

Zaproponuj strukturę danych umożliwiającą wydajne wykonywanie powyższych operacji.

*Zadanie 1, kolokwium 2, 2019/20*

Zaprojektuj strukturę danych, która umożliwia wydajne wykonywanie następujących operacji na dynamicznym ciągu liczbowym  $C$ :

- $\text{Init}::$  utwórz pusty ciąg  $C$  (operacja wykonywana tylko raz na samym początku),
- $\text{Insert}(i, x)::$  wstaw element  $x$  jako  $i$ -ty element ciągu  $C$  (za elementem z dotychczasowej pozycji  $i - 1$  a przed dotychczasowym elementem z pozycji  $i$ ),  $1 \leq i \leq |C| + 1$ ,
- $\text{Delete}(i)::$  usuń element z pozycji  $i$  z ciągu  $C$ ,  $1 \leq i \leq |C|$ ,
- $\text{Element}(i)::$  podaj wartość elementu z pozycji  $i$  w ciągu  $C$ ,  $1 \leq i \leq |C|$ ,
- $\text{isConstant}(i, j)::$  sprawdź, czy w podciągu  $C[i..j]$  wszystkie elementy są takie same,  $1 \leq i \leq j \leq |C|$ ,
- $\text{mostCommon}::$  podaj wartość najczęściej pojawiającego się elementu w ciągu  $C$  (w przypadku kilku takich elementów wystarczy podać wartość tylko jednego z nich).

*Zadanie 4, egzamin, 2018/19*

Niech  $a[1..n+2]$  będzie tablicą liczb całkowitych z przedziału  $[1..n]$ , dla pewnego dodatniego  $n$ . Grafem  $G_a$  nazywamy nieskierowany graf  $(V, E)$ , w którym zbiory wierzchołków i krawędzi są zdefiniowane jako  $V = \{1, 2, \dots, n\}$  oraz

$$E = \{i - j : i \neq j \text{ oraz } \{a[i], a[i + 1], a[i + 2]\} \cap \{a[j], a[j + 1], a[j + 2]\} \neq \emptyset\}.$$

- (a) Jaka może być maksymalna wysokość DFS-drzewa w grafie  $G_a$  o korzeniu w wierzchołku 1?

- (b) Jaka może być maksymalna wysokość BFS-drzewa w grafie  $G_a$  o korzeniu w wierzchołku 1?
- (c) Udowodnij, że graf  $G_a$  jest dwuspójny wierzchołkowo (jest spójny i nie zawiera wierzchołków rozdzielających).
- (d) Przyjmijmy, że wagą krawędzi  $i - j$  jest wartość najmniejszego elementu w zbiorze  $\{a[i], a[i + 1], a[i + 2]\} \cap \{a[j], a[j + 1], a[j + 2]\}$ . Zaprojektuj efektywny algorytm, który dla danej tablicy  $a$  oblicza wagę najlżejszego drzewa rozpinającego dla grafu  $G_a$ .

Uzasadnij poprawność swoich rozwiązań i przeprowadź analizę złożoności obliczeniowej zaproponowanych algorytmów.