

ZMP - Lista 6

Marcin Michalski, WMAT PWr

Maj 2024

Zadania tradycyjnie należy rozwiązać zgodnie ze specyfikacją i udostępnić prowadzącemu laboratoria.

Deadline: 11.05.2024, 23:59

Ćwiczenie 1. Powtórz wybrane ćwiczenia z poprzedniej listy, tym razem używając klas zamiast struktur.

Ćwiczenie 2. Niech graf $T = (V, E, r, f)$ będzie drzewem binarnym z wyróżnionym wierzchołkiem $r \in V$ (korzeniem) i funkcją "kierunku" $f : V \setminus \{r\} \rightarrow \{0, 1\}$. Załóżmy, że $(x, y) \in E^1$ oraz $f(y) = 0$. Rotacją w prawo wokół krawędzi (x, y) nazywamy operację, wskutek której z drzewa T otrzymujemy drzewo $T' = (V, E', r', f')$, gdzie

$$r' = \begin{cases} r, & \text{gdy } x \neq r; \\ y, & \text{gdy } x = r; \end{cases}$$
$$E' = E \setminus \{(p_T(x), x), (x, y), (y, r_T(y))\} \cup \{(p_T(x), y), (y, x), (x, r_T(y))\},$$
$$f' = f \setminus \{(x, f(x)), (y, 0), (r_T(y), 1)\} \cup \{(x, 1), (y, f(x)), (r_T(y), 0)\},$$

gdzie $l_T(v), r_T(v), p_T(v)$ oznaczają funkcje zwracające kolejno lewego potomka, prawego potomka, rodzica wierzchołka v w drzewie T^2 .

- Prześledź, co robi rotacja w prawo wokół danej krawędzi, na podstawie komunikatywnego rysunku.
- Pokaż, że po rotacji w prawo BST (patrz Zadanie 2) pozostaje BST.
- Zdefiniuj analogiczną operację rotacji w lewo i zaagituj dla niej analogiczny fakt.
- Zaimplementuj obie te rotacje w C++.

Rozwiązania zadań umieść w 16/zi, gdzie i, to nr zadania, w swoim repozytorium. Pliki nazwij <nr_indeksu>izic.cpp, gdzie i, to nr zadania, a znak c to podpunkt (jeśli zadanie nie ma podpunktów, to c należy pominąć). Przykład: 123456z1a.cpp.

¹Dla wygody zakładamy, że graf jest skierowany, tutaj strzałki są od rodziców do dzieci.

²Założmy też, że funkcje te są określone wtedy, kiedy to możliwe, tzn. dla liści nie są określone l_t i r_t , a dla korzenia p_T . Należy to mieć na uwadze przy definicjach E' oraz f' , np. jeśli $x = r$, to pary z $p_T(x)$ należy pominąć. Dla maksimum precyzji definicje E' i f' trzeba by napisać z 4 przypadkami, ale czytelność by wtedy znacznie ucierpiała.

Zadanie 1 (5p.). Zaimplementuj klasę `Vector3d`, która będzie reprezentować wektory w przestrzeni \mathbb{R}^3 . Klasa powinna składać się z pól oznaczających współrzędne wektora oraz metod do wyświetlania, dodawania, mnożenia przez skalar, obliczania iloczynu skalarnego, oraz sprawdzania równoległości i prostopadłości wektorów.

Na potrzeby testów założmy, że program po uruchomieniu będzie się spodziewał ze standardowego wejścia 7 liczb rzeczywistych $v_1, v_2, v_3, w_1, w_2, w_3, a$ oznaczających kolejne współrzędne dwóch wektorów i skalar, a zwróci $av, v + w, v \circ w$, oraz czy $v \parallel w$ i czy $v \perp w$.

Zadanie 2 (5p.). Zaimplementuj za pomocą klas binarne drzewo poszukiwań (BST - binary search tree), tzn. drzewo binarne, dla którego wartość w każdym węźle jest nie mniejsza od wartości w lewym potomku i nie większa od wartości w prawym potomku.

- (a) Wykorzystaj je do posortowania ciągu liczb całkowitych. Program powinien spodziewać się ze standardowego wejścia liczby naturalnej n , a następnie n liczb całkowitych do posortowania. Następnie powinien zwrócić je posortowane.
- (b) * Uodpornij swój program na złośliwe przykłady danych poprzez użycie samobalansującego BST³.

³https://en.wikipedia.org/wiki/Self-balancing_binary_search_tree