Drzewa poszukiwań binarnych (BST)

Maciej Hojda

Uwaga: Słowa „dany", „zadany", „podany", „wybrany" itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

1 Zadanie nr 1 – weryfikacja, konstrukcja

- Zaimplementuj strukturę danych do przechowywania takiego grafu, gdzie każdy wierzchołek posiada najwyżej trzech sąsiadów: poprzednika („ojca") i dwóch następników („synów") – lewego i prawego.
- Zaimplementuj procedurę sprawdzania, czy zadany graf spełnia warunek BST, tzn. jest drzewem i dla każdego wierzchołka: wszystkie wierzchołki po stronie lewego następnika (włącznie) mają wartości mniejszą od wartości tego wierzchołka (dla prawego następnika analogicznie – wartości większe).
- Zaimplementuj funkcję graficznie wyświetlającą przechowywany graf.
- Zaimplementuj metodę korekty pozycji wierzchołka, który nie spełnia warunku BST.
- Zaimplementuj metodę tworzenia zrównoważonego (o możliwie najmniejszej głębokości) drzewa przeszukiwania binarnego z zadanego zbioru wierzchołków (wartości). Wykorzystaj wybraną metodę sortowania wraz z podejściem dziel i zwyciężaj.

2 Zadanie nr 2 – słownik

- Zaimplementuj metodę generującą drzewo przeszukiwania binarnego na podstawie struktury robotów z wcześniejszych list. Kluczowy parametr struktury (parametr robota, względem którego tworzone jest drzewo) zadaje użytkownik.
- Zaimplementuj efektywne algorytmy:
  - przechodzenia drzewa metodami inorder, preorder i postorder,
  - wyszukiwania elementu (i ścieżki prowadzącej do tego elementu, zaczynając od korzenia),
  - wyszukiwania minimum i maksimum,
  - wyszukiwania następnika i poprzednika zadanego węzła,
  - wstawiania i usuwania węzła.

3 Zadanie nr 3 – drzewa czerwono-czarne

- Zaimplementuj strukturę danych do przechowywania drzewa czerowno-czarnego (drzewa przeszukiwań binarnych o dodatkowym polu – kolorze – w każdym wierzchołku).
- Zaimplementuj algorytm weryfikacji, czy zadane drzewo spełnia własność czerwono-czarną (każdy liść jest czarny, jeśli węzeł jest czerwony, to jego synowie są czarni, każda ścieżka prosta z ustalonego węzła do liścia ma tyle samo czarnych węzłów).
- Zaimplementuj efektywne algorytmy:
  - prawej i lewej rotacji,
    Prawa rotacja dwóch węzłów – **ojca** i **prawego syna** – polega na zamianie węzła **ojca** na węzeł **prawego syna**, a prawego syna ojca na lewego syna **prawego syna**. Lewa rotacja analogicznie.
  - wstawiania i usuwania węzła (przy zachowaniu własności drzewa czerwono-czarnego).