

Struktury drzewiaste

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

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

1. Zaimplementuj strukturę danych do przechowywania drzewa binarnego (każdy wierzchołek ma najwyżej dwóch potomków).
2. Zaimplementuj funkcję dodającą i usuwającą wierzchołek.
3. Zaimplementuj funkcję graficznie wyświetlającą przechowywany graf.
4. Zaimplementuj funkcję zapisującą drzewo do pliku i wczytującą drzewo z pliku. Zaproponuj format pliku.

Wejście: plik z drzewem.

Wyjście: wyświetlone drzewo; przykład dodawania i usuwania węzłów.

2 Zadanie nr 2 – słownik

1. Zaimplementuj metodę generującą drzewo poszukiwań binarnych na podstawie struktury robotów z wcześniejszych list. Kluczowy parametr struktury (parametr robota, względem którego tworzone jest drzewo) zadaje użytkownik.
2. Zaimplementuj algorytmy dodawania, usuwania i wyszukiwania węzłów.
3. Zaimplementuj funkcję graficznie wyświetlającą przechowywany graf.
4. Zaimplementuj funkcję zapisującą drzewo do pliku i wczytującą drzewo z pliku. Zaproponuj format pliku.

Wejście: lista robotów (wczytywana z pliku).

Wyjście: drzewo BST (wizualizacja); przykład dodawania i usuwania węzłów.

3 Zadanie nr 3 – drzewa R-B, rotacja

1. Zaimplementuj strukturę danych do przechowywania drzewa czerowno-czarnego na podstawie struktury robotów z wcześniejszych list. Kluczowy parametr struktury (parametr robota, względem którego tworzone jest drzewo) zadaje użytkownik.

2. Zaimplementuj algorytmy dodawania, usuwania i wyszukiwania węzłów.
3. Zaimplementuj funkcję graficznie wyświetlającą przechowywany graf.
4. Zaimplementuj funkcję zapisującą drzewo do pliku i wczytującą drzewo z pliku.
5. Zaimplementuj algorytmy prawej i lewej rotacji w drzewie czerwono-czarnym. Zaproponuj format pliku.

Wejście: lista robotów (wczytywana z pliku).

Wyjście: drzewo R-B (wizualizacja), przykład rotacji lewej i prawej.

4 Zadanie nr 4 – drzewa R-B, dodawanie/usuwanie

Rozwiń zadanie nr 3.

1. 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).
2. Zaimplementuj algorytmy wstawiania i usuwania węzła.

Wejście: drzewo czerwono-czarne (wczytywane z pliku).

Wyjście: drzewo R-B (wizualizacja), przykład dodawania i usuwania węzłów.

5 Zadanie nr 5 – przeszukiwanie

Rozwiń zadanie nr 4.

Zaimplementuj algorytmy przechodzenia drzewa metodami inorder, preorder i postorder.

Procedury przedstaw graficznie, krok po kroku.

Wejście: drzewo czerwono-czarne (wczytywane z pliku).

Wyjście: wizualizacja algorytmów przechodzenia drzewa.