

ALGORYTMY I STRUKTURY DANYCH

DRZEWA BST

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

1. **[**]** Wylicz ile jest różnych drzew binarnych o n węzłach. Ułóż odpowiednie równanie rekurencyjne. Czy wiesz jakie jest jego rozwiązanie?
2. **[*]** Pokaż, jak za pomocą kolejki można wypisać zawartość drzewa binarnego poziomami (korzeń znajduje się na poziomie 0, jego synowie na poziomie 1, wnukowie korzenia na poziomie 2, itd).
3. **[**]** Niech dane będzie drzewo binarne o n węzłach. W porządku *preorder* węzły te tworzą ciąg $u_1, u_2, \dots, u_n$, a w porządku *inorder* ciąg $u_{\sigma_1}, u_{\sigma_2}, \dots, u_{\sigma_n}$. Pokaż, jak za pomocą stosu można zrealizować permutację σ (przekształcić ciąg $1, 2, \dots, n$ w ciąg $\sigma_1, \sigma_2, \dots, \sigma_n$)?
4. **[**]** Przedstaw algorytm, który podzieli drzewo BST na dwa odrębne drzewa BST względem zadanej wartości klucza x . W jednym z wynikowych drzew mają się znaleźć wszystkie węzły pierwotnego BST z kluczami $\leq x$, a w drugim z kluczami $> x$. Czas działania twojego algorytmu powinien być ograniczony przez głębokość węzła z wartością x w pierwotnym drzewie.
5. **[*]** Napisz w pseudokodzie procedury, które znajdują element poprzedni i następny co do wielkości względem zadanej wartości klucza x w drzewie BST.
6. **[***]** Procedura szukająca klucza x w drzewie BST *search(x)* odwiedza wszystkie węzły od korzenia aż do węzła z wartością x (lub gdy x nie ma w drzewie, to do węzła z wartością x' , najbliższą x z lewej albo z prawej strony). Wypisanie węzłów leżących na ścieżce od korzenia do x (albo do x') jest więc trywialnym zadaniem. Przedstaw algorytm, który pozwoli wypisać węzły w odwrotnej kolejności — od x (albo od x') do korzenia. Twój algorytm ma działać w miejscu w czasie proporcjonalnym do głębokości węzła z wartością x (albo z wartością x'). Nie wolno przy tym zniszczyć pierwotnej struktury drzewa. Czy potrafisz to zadanie rozwiązać w przypadku, gdy klucze w drzewie mogą się powtarzać?
7. **[**]** Niech T będzie n -elementowym drzewem SPLAY, w którym zapamiętane są kolejne liczby naturalne $1, \dots, n$. Przedstaw taki ciąg operacji *splay()*, w wyniku którego otrzymamy drzewo o wysokości $n - 1$.
8. **[**]** Niech T będzie drzewem SPLAY, w którym na głębokości h znajduje się węzeł z kluczem x . Po wykonaniu operacji *splay(x)* na drzewie T otrzymamy drzewo T' z wartością x w korzeniu. Udowodnij, że wszystkie węzły leżące na ścieżce z korzenia do x w drzewie T

po wykonaniu operacji $splay(x)$ zmniejszą swoją głębokość o połowę w nowym drzewie T' .
Ściślej, jeśli węzeł y leżał w drzewie T na ścieżce z korzenia do x na głębokości h_y , to po wykonaniu operacji $splay(x)$ znajdzie się w nowym drzewie T' na głębokości $h'_y \leq \frac{h_y}{2} + 2$.