Ćwiczenia: lista zadań nr 15 *17 stycznia 2006*

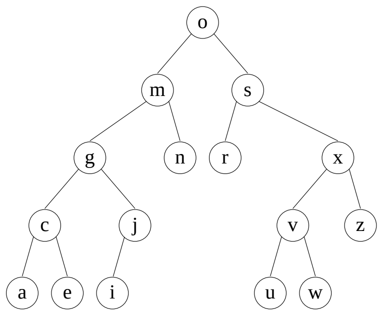
Algorytmy i Struktury Danych

drzewa zrównoważone

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. [∗] Przedstaw implementacje pojedynczych rotacji w lewo i w prawo. Jak wykorzystać te operacje do implementacji podwójnych rotacji?
2. [∗∗] Oblicz, jaka jest maksymalna różnica pomiędzy liczbą węzłów w lewym i w prawym poddrzewie drzewa AVL o wysokości h (wylicz tę różnicę w korzeniu).
3. [∗∗] Jaka jest maksymalna ilość rotacji wykonywanych w drzewie AVL podczas wstawiania (operacja *insert()*) a jaka ilość podczas usuwania (operacja *delete()*)? Co jest przyczyną tak istotnej różnicy?
4. [∗∗] Drzewo Fibonacciego jest minimalnym drzewem AVL. Czy jest ono również drzewem R–B (jeśli tak, to pokoloruj jego węzły)?
5. [∗∗] Oblicz, jaka jest maksymalna różnica pomiędzy liczbą węzłów w lewym i w prawym poddrzewie drzewa R–B o wysokości h (wylicz tę różnicę w korzeniu).
6. [∗∗∗] Udowodnij, że drzewo R–B o n węzłach ma wysokość co najwyżej $2\log(n + 2) - 3$. ***Wskazówka:*** Wyznacz liczbę węzłów w minimalnym drzewie czerwono–czarnym o zadanej wysokości (rozpatrz osobno przypadki dla wysokości parzystych i nieparzystych). W drzewie takim czerwone węzły występują tylko na jednej ścieżce z korzenia do liścia, na ścieżce tej co drugi węzeł jest czerwony i liść jest czerwony. Wynika to z konstrukcji minimalnego drzewa czerwono–czarnego o wysokości h, w którym po jednej stronie korzenia znajduje się minimalne drzewo czerwono–czarne o wysokości $h - 1$ i czarnej wysokości h_b a po drugiej drzewo pełne o wysokości h_b z samymi czarnymi węzłami.
7. [∗∗∗] Opisz algorytm, według którego usuwany jest element z drzewa R–B, a następnie przedstaw rekurencyjną implementację tego algorytmu.
8. Na przedstawionym poniżej drzewie SPLAY wykonaj kolejno operacje:
   (a) [∗] *search(g)*;
   (b) [∗] *insert(y)*;
   (c) [∗] *delete(j)*.

   Zilustruj, jak będzie wyglądało to drzewo po wykonaniu każdej operacji (i po każdym kroku rozchylającym w operacji *splay()*).



9. [$**$] Niech T będzie n-elementowym drzewem SPLAY, w którym pamiętane są kolejne liczby naturalne $1, \ldots, n$. Przedstaw taki ciąg operacji *splay()* na drzewie T, w wyniku którego otrzymamy drzewo o wysokości $n-1$.

10. [$***$] Niech T będzie drzewem SPLAY, w którym na głębokości h znajduje się węzeł z kluczem x. Po wykonaniu operacji *T.splay(x)* otrzymujemy drzewo SPLAY T' z korzeniem w węźle x. Udowodnij, że wszystkie węzły leżące na ścieżce z korzenia do x w drzewie T po wykonaniu operacji *splay()* zmiejszą 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 *T.splay(x)* znajdzie się w nowym drzewie T' na głębokości $h'_y \leq \frac{h_y}{2} + 2$.