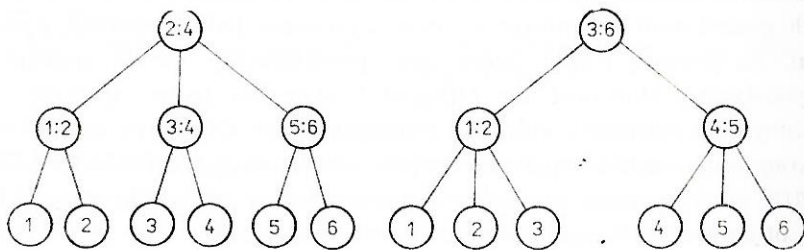


postaci drzewa podczas wykonywania ciągu instrukcji INSERT i DELETE. Na przykład, jeżeli wykonujemy ciąg instrukcji DELETE, które usuwają wierzchołki tylko z lewej strony drzewa, to bez okresowego przywracania drzewu zrównoważonej postaci, otrzymalibyśmy drzewo niezrównoważone.

Znanych jest kilka sposobów utrzymywania drzew w postaci zrównoważonej. Niektóre z nich pozostawiają strukturze drzewa sporą dowolność, np. liczba wierzchołków w drzewie o wysokości h może być dowolną liczbą z przedziału od 2^h do 2^{h+1} lub 3^h . Metody takie pozwalają do co najmniej dwukrotnego zwiększenia liczby wierzchołków w poddrzewie, zanim zajdzie potrzeba dokonania zmian w tej części drzewa, która znajduje się powyżej korzenia danego poddrzewa.

Omówimy dwie metody tego typu oparte odpowiednio o 2 - 3 drzewa i AVL drzewa. Ponieważ algorytmy działające na 2 - 3 drzewach są łatwiejsze do zrozumienia, więc w tekście omówimy właśnie te algorytmy. Natomiast algorytmy oparte o AVL drzewa, które są zresztą podobne do algorytmów dla 2 - 3 drzew, przenosimy do ćwiczeń.

DEFINICJA. 2 - 3 drzewem nazywamy drzewo, w którym każdy wierzchołek nie będący liściem ma 2 lub 3 następniki oraz każda ścieżka od korzenia do liścia ma tę samą długość. Na przykład, drzewo złożone



Rys. 4.26. 2 - 3 drzewa

z jednego wierzchołka jest 2 - 3 drzewem. Rysunek 4.26 przedstawia dwa 2 - 3 drzewa o sześciu liściach.

Następujący lemat wiąże liczbę wierzchołków i liści 2 - 3 drzewa z jego wysokością.

LEMAT 4.6. Niech T będzie dowolnym 2 - 3 drzewem o wysokości h . Liczba wierzchołków drzewa T należy do przedziału między $2^{h+1} - 1$ i $(3^{h+1} - 1)/2$, a liczba liści do przedziału między 2^h i 3^h .

Dowód. Zwykła indukcja względem h . $\square$

Linioowo uporządkowany zbiór S można reprezentować za pomocą 2 - 3 drzewa w ten sposób, że elementy zbioru S przyporządkowujemy liściom drzewa. Element przyporządkowany liściowi l oznaczamy przez $E[l]$. Istnieją dwie podstawowe metody przyporządkowywania elementów liściom; której z nich należy użyć, zależy od rozważanego zastosowania.

Jeżeli uniwersum możliwych elementów jest znacznie większe niż liczba faktycznie używanych elementów oraz drzewo ma służyć jako słownik, to wówczas przeważnie wiążemy elementy z liśćmi w porządku rosnącym od lewej strony drzewa do prawej. W każdym wierzchołku v , który nie jest liściem, zapisujemy dwie dodatkowe informacje, $L[v]$ i $M[v]$. $L[v]$ jest największym elementem zbioru S w poddrzewie, którego korzeniem jest lewy następnik wierzchołka v ; $M[v]$ jest największym elementem zbioru S w poddrzewie, którego korzeniem jest drugi (od strony lewej) następnik wierzchołka v (zob. rysunek 4.26).⁽¹⁾ Wartości L i M , które są przyporządkowane wierzchołkom, umożliwiają poszukiwanie elementu w drzewie w sposób analogiczny jak przy poszukiwaniu binarnym (poszukiwanie rozpoczyna się oczywiście od korzenia). Czas potrzebny do znalezienia elementu jest proporcjonalny do wysokości drzewa. Zatem, jeżeli dany n -elementowy zbiór reprezentujemy w opisany sposób jako 2 - 3 drzewo, to wówczas instrukcję MEMBER można wykonać w czasie $O(\log n)$.

Natomiast przy drugiej metodzie przyporządkowywania elementów liściom, na porządek, w jakim te elementy są przyporządkowywane nie nakłada się żadnych ograniczeń. Metoda ta jest szczególnie przydatna do implementacji instrukcji UNION. Jednak, aby móc wykonać instrukcje takie jak DELETE, trzeba wprowadzić dodatkowy mechanizm pozwalający znajdować liść reprezentujący dany element. Jeżeli elementami zbiorów są

⁽¹⁾ Przeważnie modyfikuje się nieco znaczenie atrybutu $M[v]$. Mianowicie, przyjmuje się, że $M[v]$ jest niezdefiniowane, jeżeli wierzchołek v ma tylko dwa następniki. Jeżeli wierzchołek v ma trzy następniki, to wtedy za $M[v]$ przyjmuje się wartość zdefiniowaną w tekście. Upraszcza to trochę modyfikację atrybutów L i M podczas wstawiania i usuwania elementu (przyj. tłum.).