



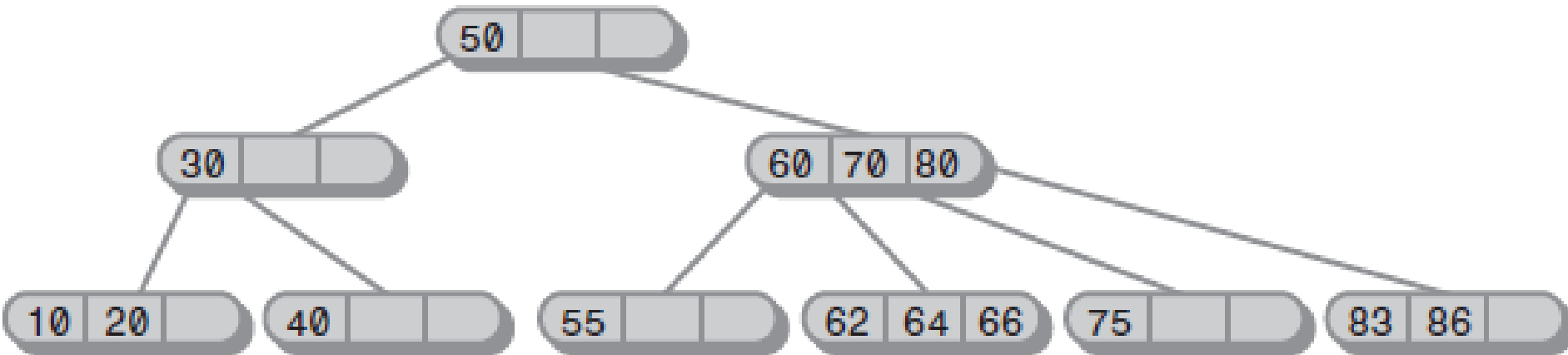
Universitatea Politehnica din București
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare



ARBORI 2-3-4

Arbori 2-3-4

- Arborii 2-3-4 sunt arbori multicăi, în care fiecare nod poate avea cel mult patru fii
- Arborii 2-3-4 sunt arbori echilibrați, la fel ca arborii bicolori
- Eficiența lor este ceva mai scăzută, dar sunt mai ușor de programat



Observații

- Fiecare **nod** poate conține **unul, două sau trei elemente**
- Primele trei noduri au fii, în timp ce toate cele șase noduri de pe ultimul nivel sunt frunze, care prin definiție nu au fii
- Într-un arbore 2-3-4, **toate frunzele se află întotdeauna pe același nivel**

Explicația numelui

- Cifrele 2,3 și 4 se referă la numărul legăturilor către noduri fii, care pot fi conținute de un nod oarecare
- Pentru nodurile care nu sunt frunze, sunt posibile 3 situații:
 - Un nod cu 1 element are întotdeauna 2 fii
 - Un nod cu 2 elemente are întotdeauna 3 fii
 - Un nod cu 3 elemente are întotdeauna 4 fii

Observații

- Un nod care nu este frunză trebuie să aibă cu un fiu mai mult decât numărul de elemente pe care îl conține
- Dacă numărul legăturilor către fii este **L**, iar numărul elementelor este **D**, avem:
- **$L = D + 1$**

Observații

- Un nod frunză nu are niciun fiu, dar poate conține una, două sau chiar trei elemente
- Nu se admite existența nodurilor vide
- Din cauză că arborii 2-3-4 pot avea noduri cu până la patru fii, ei se mai numesc **arbori multicăi de ordinul al patrulea**

Observații

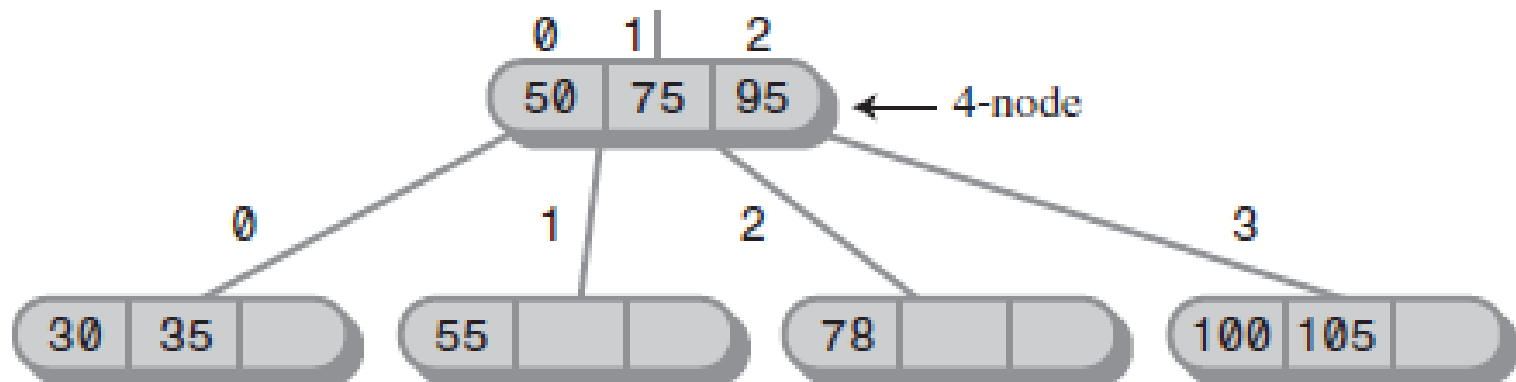
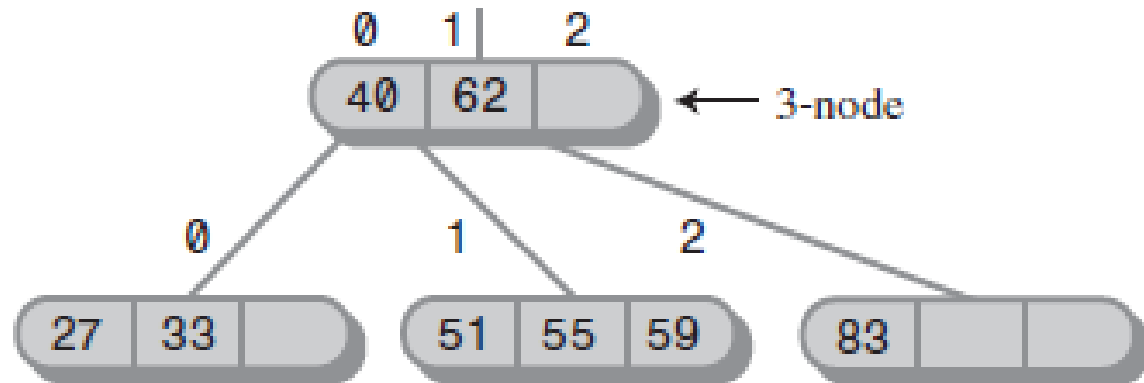
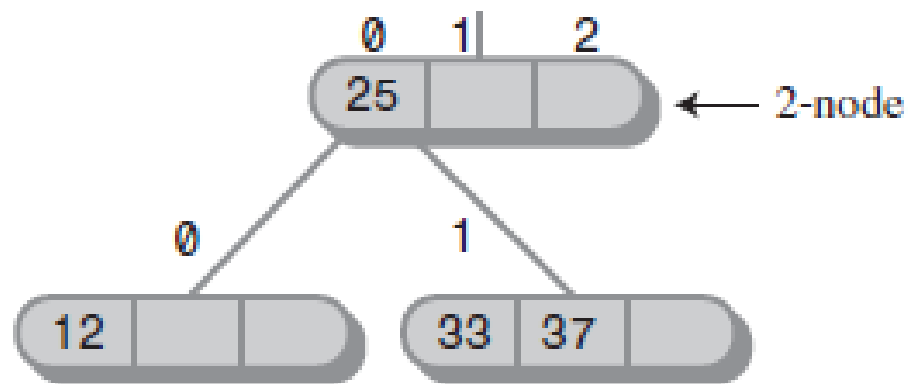
- Un nod nu poate avea un singur fiu, așa cum se întâmplă cu nodurile din arborii binari
- Un arbore binar poate fi gândit ca un caz particular de arbore multicăi de ordinul al doilea, din cauză că fiecare nod poate avea maximum doi fii

Observații

- Într-un arbore binar, un nod are cel mult doi fii
- Situația în care un nod are o singură legătură, către fiul său drept, sau către cel stâng, este permisă
- Cealaltă legatură poate avea valoarea NULL

Observații

- În arborii 2-3-4, existența nodurilor cu un singur fiu nu este permisă
- Un nod cu un singur element trebuie să aibă două legături, cu excepția cazului în care nodul este o frunză și nu are nicio legătură
- Un nod cu 2 legături se numește **2-nod**, cu 3 legături **3-nod**, iar cu 4 legături **4-nod**

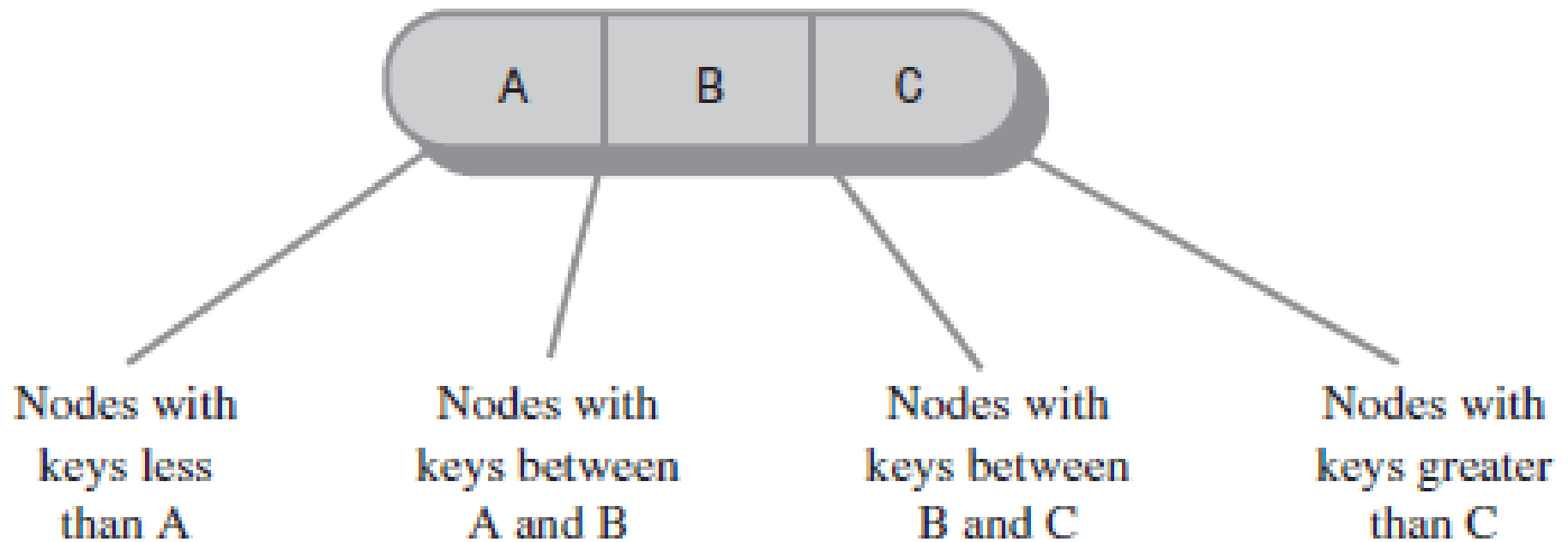


Convenții

- Se numerotează elementele dintr-un nod cu indici de la 0 la 2, iar legăturile către fii, cu indici de la 0 la 3
- Elementele dintr-un nod sunt dispuse în **ordinea crescătoare** a cheilor
- Elementele se scriu de la stânga la dreapta (de la indici mici spre indici mari)

Reguli

- Toți fiii din subarborele cu rădăcina în fiul 0 au chei mai mici decât elementul 0
- Toți fiii din subarborele cu rădăcina în fiul 1 au chei mai mari decât elementul 0, dar mai mici decât elementul 1
- Toți fiii din subarborele cu rădăcina în fiul 2 au chei mai mari decât elementul 1, dar mai mici decât elementul 2
- Toți fiii din subarborele cu rădăcina în fiul 3 au chei mai mari decât elementul 2



Observații

- Echilibrul arborelui se păstrează, chiar dacă se inserează o secvență ordonată crescător (sau descrescător)
- Facilitatea de autoechilibru a arborilor 2-3-4 provine din modul de inserare a noilor elemente

Căutare

- Căutarea unui element cu o anumită cheie este similară operației dintr-un arbore binar de căutare
- Se pornește de la rădăcină și, dacă nu se găsește cheia căutată, se selectează legătura care conduce la subarborele asociat domeniului corespunzător de valori

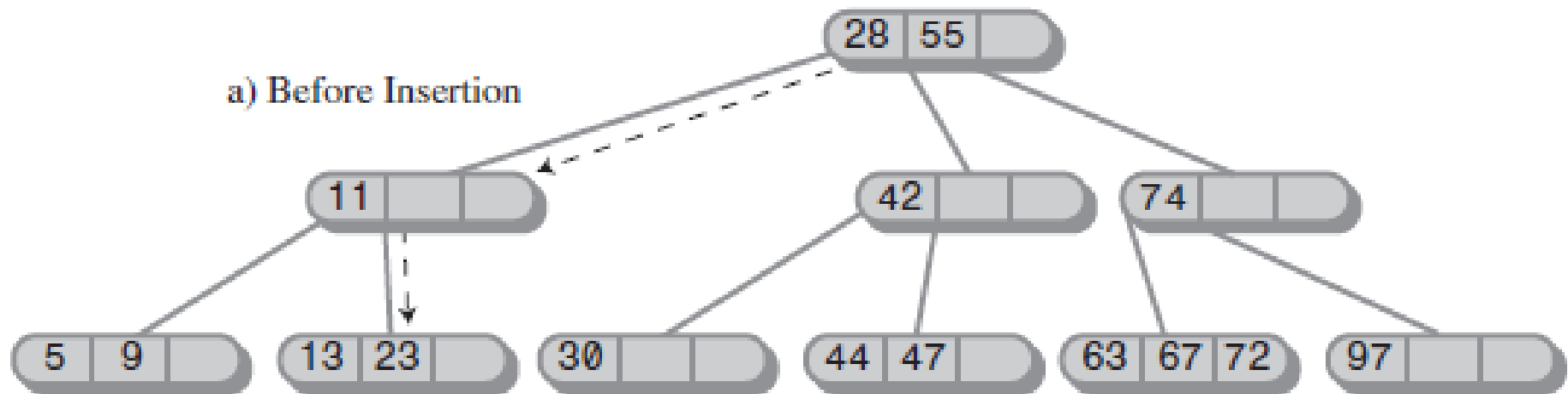
Inserare

- Noile elemente sunt întotdeauna inserate în frunze, care sunt dispuse pe nivelul de la baza arborelui
- Dacă elementele ar fi fost inserate în nodurile care au fii, atunci numărul acestor fii trebuia să se modifice, pentru a menține structura arborelui

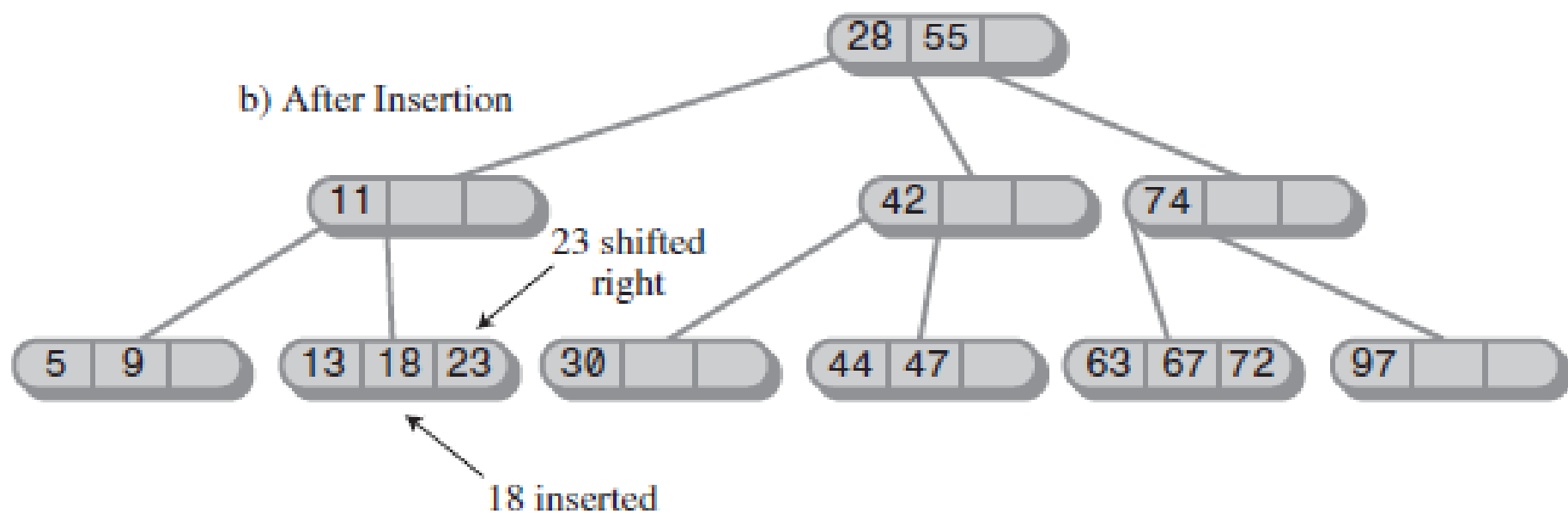
Inserare

- Dacă nu se întâlnesc noduri complete în cursul căutării, inserarea este simplă
- Când se ajunge la nodul frunză potrivit, noul element este inserat în acesta
- Inserarea poate implica deplasarea a unul sau două elemente dintr-un nod, astfel încât cheile să rămână în ordinea corectă și după inserarea noului element

a) Before Insertion



b) After Insertion



Divizarea nodurilor

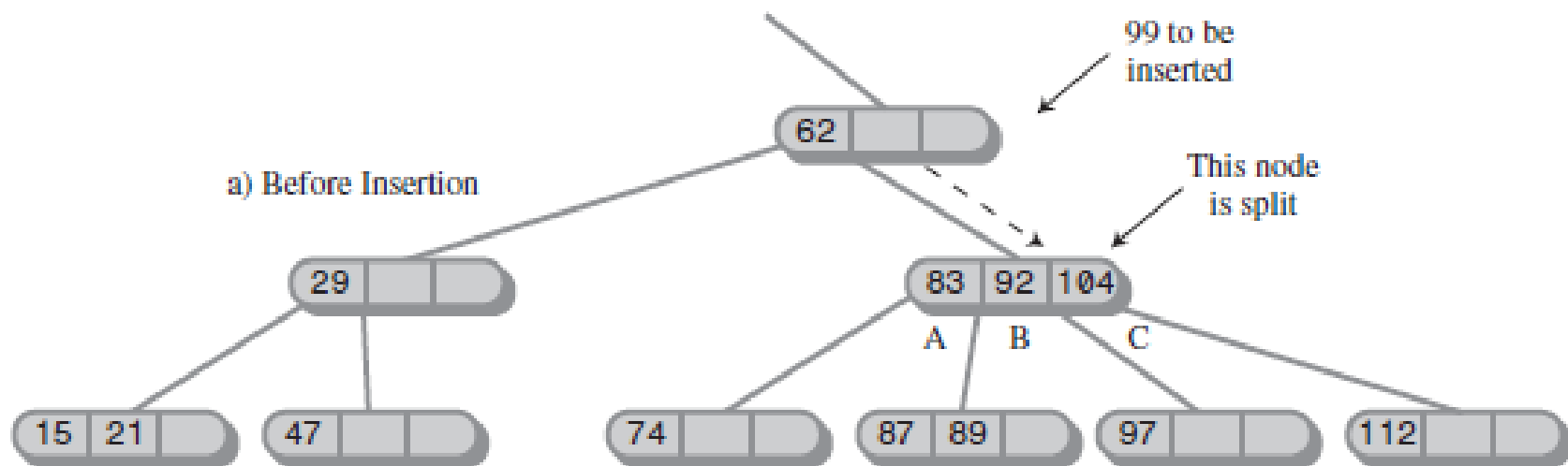
- Inserarea se complică când se întâlnește un nod complet pe calea descendentă către locul de inserare
- Dacă aceasta se întâmplă, nodul va suferi o divizare
- Rolul acestei operații este de a menține echilibrul arborelui

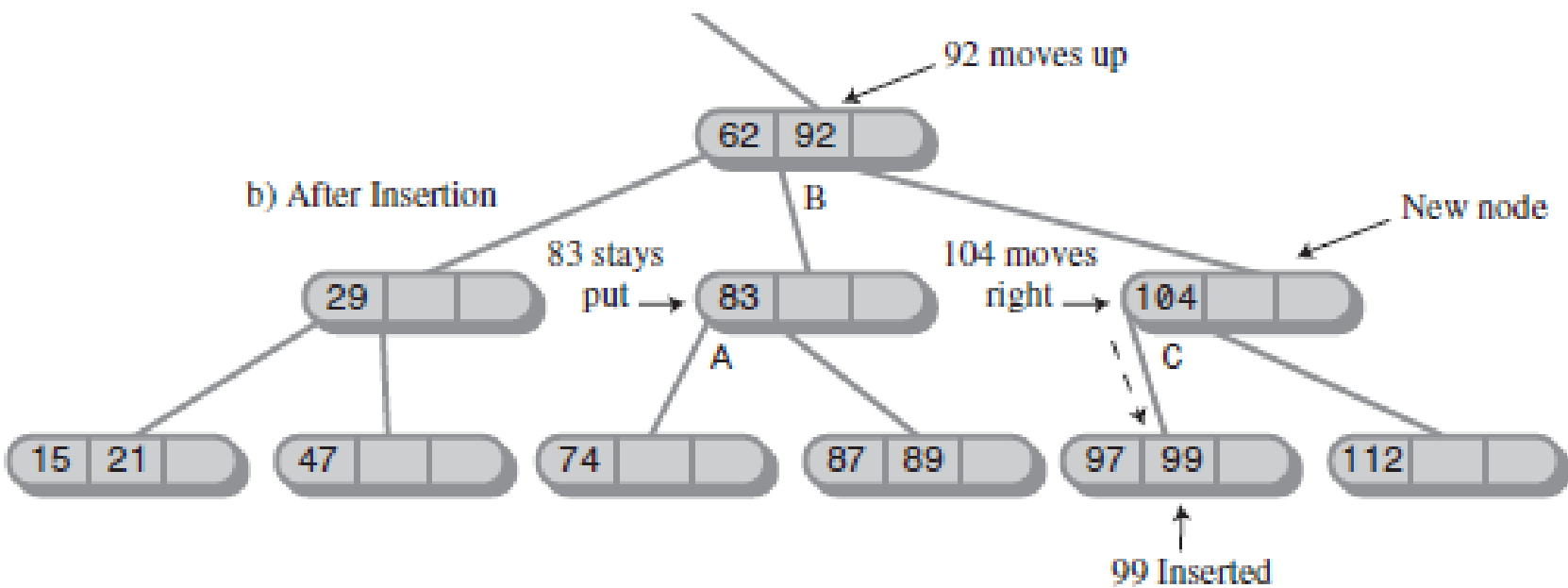
Divizarea nodurilor

- Se notează elementele din nodul care va fi divizat cu A, B și C
- Se presupune că nodul care va fi divizat nu este rădăcină
- 1. Se creează un nod nou, vid. Acesta este fratele nodului care va fi divizat, fiind plasat la dreapta acestuia

Divizarea nodurilor

- **2.** Elementul C este deplasat în noul nod
- **3.** Elementul B este deplasat în părintele nodului divizat
- **4.** Elementul A rămâne în locul în care este
- **5.** Cei doi fii situați cel mai la dreapta sunt deconectați de la nodul divizat și conectați la noul nod





Observații

- Operația de diviziune poate fi descrisă prin aceea că un 4-nod este transformat în două 2-noduri
- Efectul divizării nodului este deplasarea elementelor în sus (spre nodul părinte) și spre dreapta
- Această rearanjare menține echilibrul în arbore

Observații

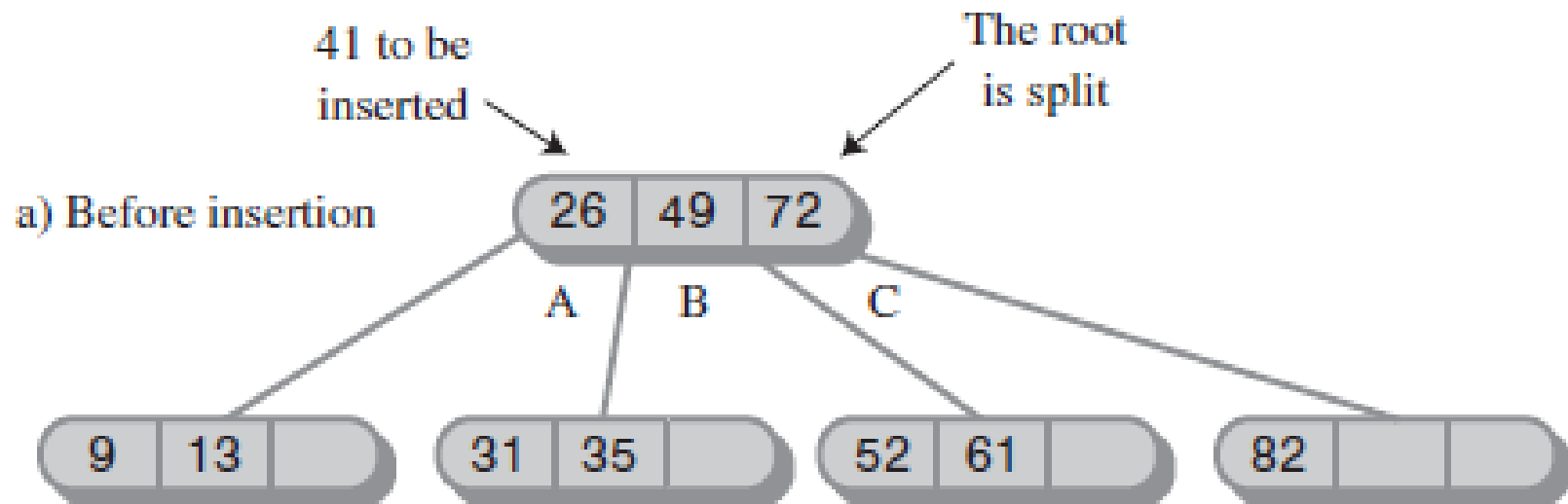
- Este posibil să existe mai multe noduri complete pe calea de la rădăcină către locul de inserare
- În acest caz, se vor efectua diviziuni multiple

Divizarea rădăcinii arborelui

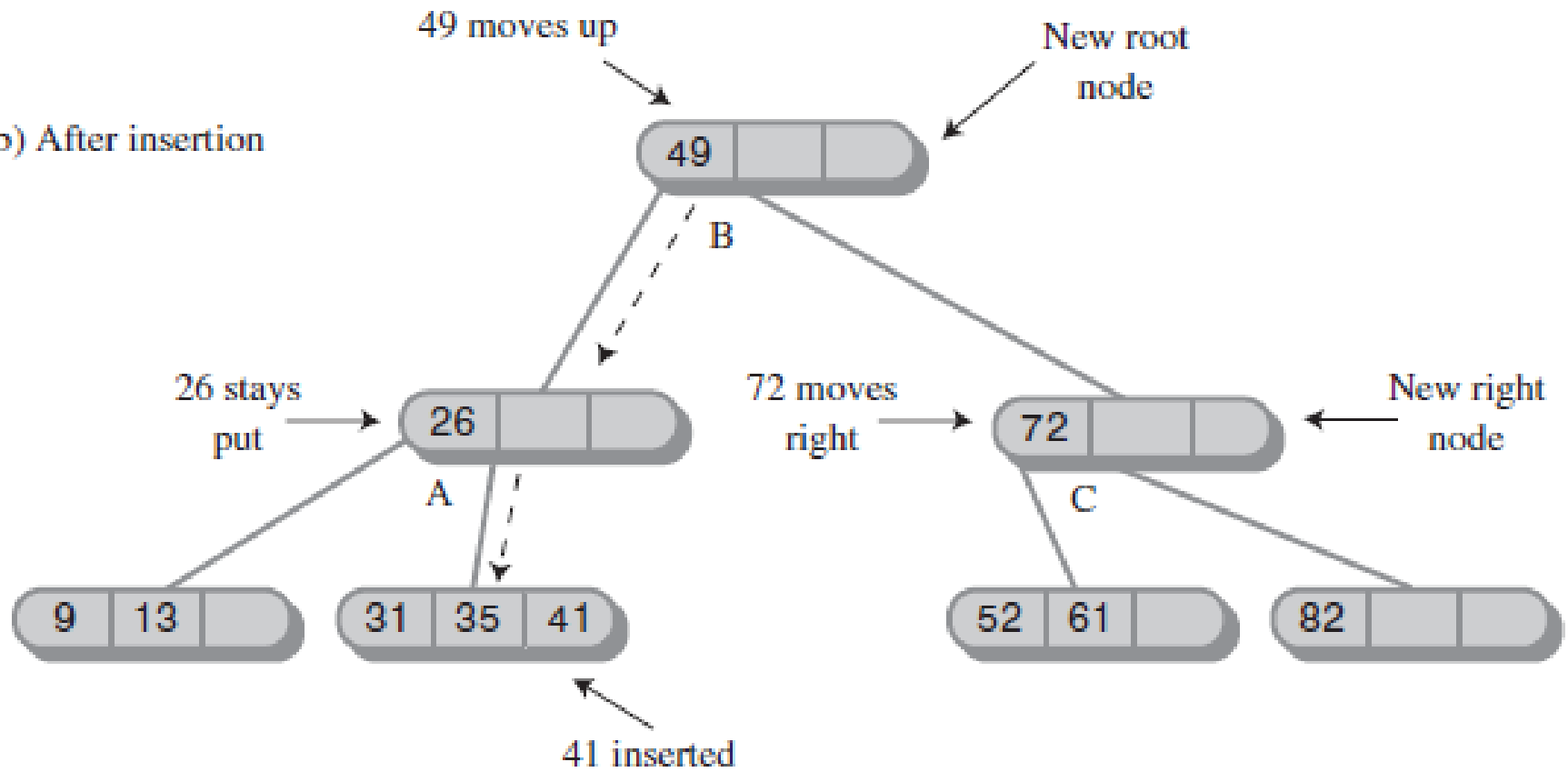
- Dacă nodul **rădăcină** întâlnit la începutul căutării locului de inserare este **complet**, procesul de divizare implică următorii pași:
- **1.** Se creează un nod nou, care va deveni noua rădăcină și părintele nodului divizat
- **2.** Se creează un al doilea nod, care va deveni fratele nodului divizat

Divizarea rădăcinii arborelui

- **3.** Elementul C este deplasat în noul frate
- **4.** Elementul B este deplasat în noua rădăcină
- **5.** Elementul A rămâne pe loc
- **6.** Cei doi fii ai nodului divizat aflați cel mai la dreapta sunt deconectați de la acel nod și conectați la noul nod aflat în partea dreaptă



b) After insertion



Observații

- Operația creează o nouă rădăcină, situată cu un nivel mai sus față de vechea rădăcină
- Înălțimea totală a arborelui crește cu o unitate
- Se poate afirma că un 4-nod a fost divizat în trei 2-noduri
- După efectuarea diviziunii, căutarea locului de inserare continuă, parcurgând arborele în jos

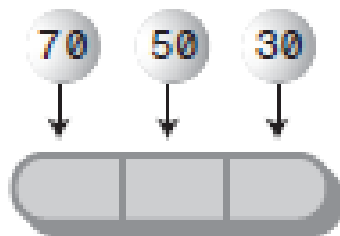
Divizarea la parcurgerea descendentă

- Din cauză că divizarea tuturor nodurilor complete are loc la parcurgerea descendentă a arborelui, efectul unei operații de divizare nu se poate propaga înapoi spre rădăcină
- Părintele unui nod care urmează să fie divizat este, cu siguranță, incomplet

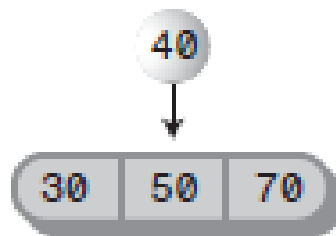
Divizarea la parcurgerea descendentă

- Elementul B din nodul divizat va putea fi inserat în nodul părinte, fără a provoca divizarea acestuia
- Dacă nodul părinte are deja doi fii înaintea divizării unuia dintre aceștia, va deveni un nod complet
- Divizarea nodului se va efectua la următoarea operație de inserare în care nodul va fi implicat

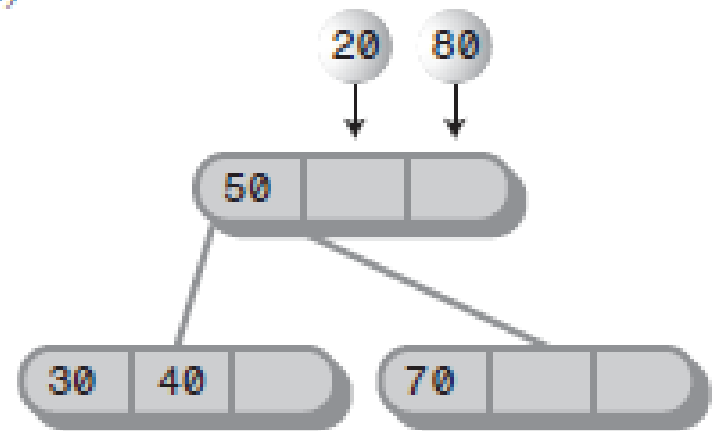
a)



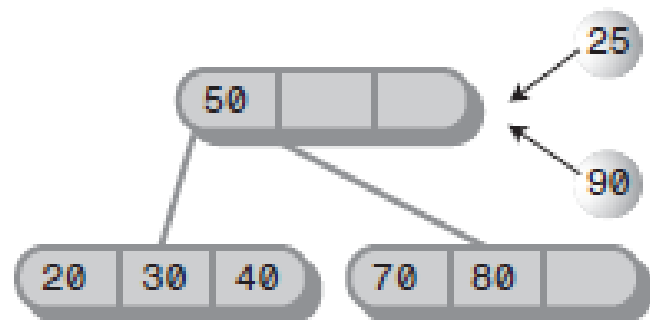
b)



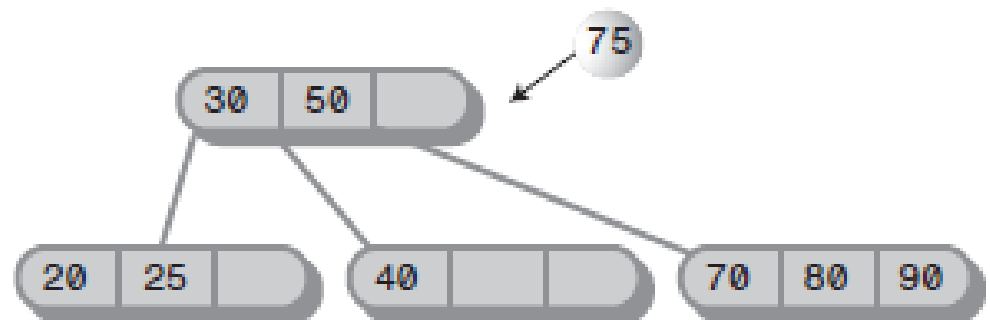
c)



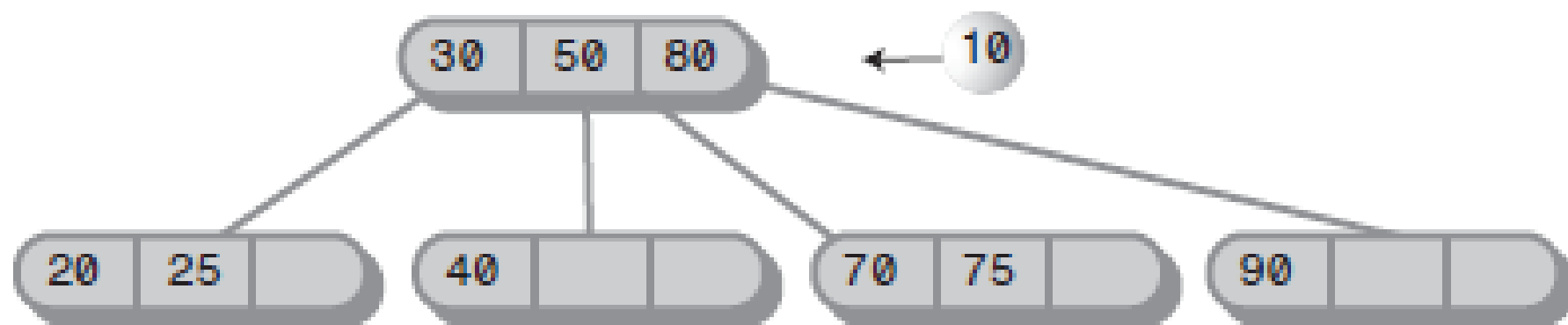
d)



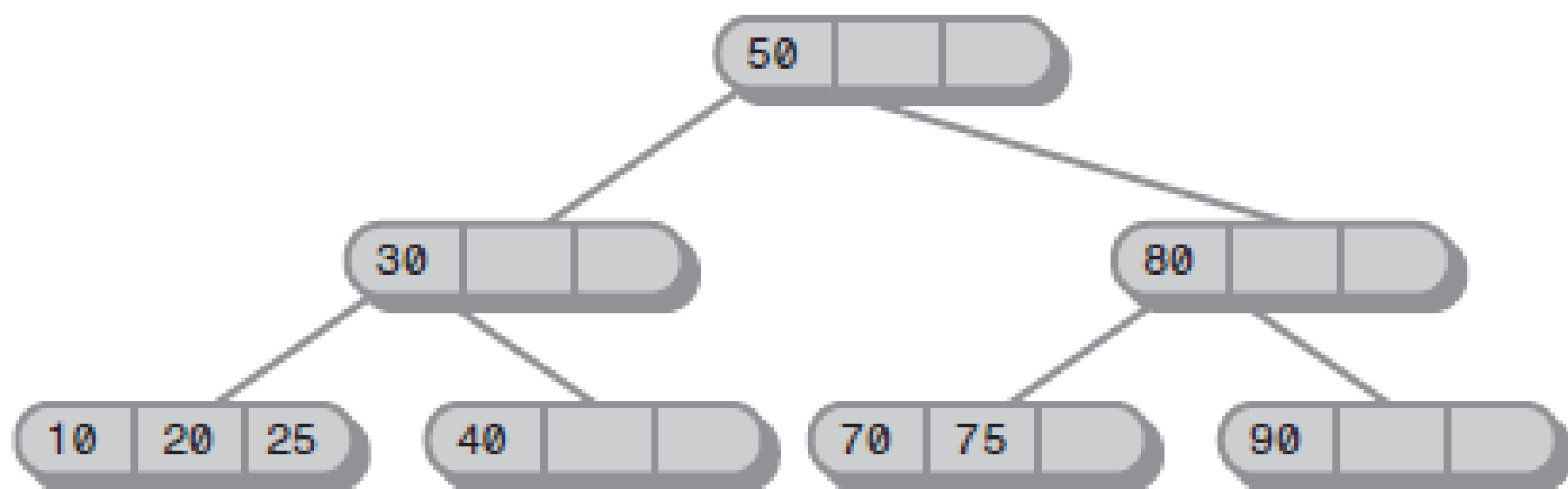
e)



f)



g)



Eficiența arborilor 2-3-4

- Analiza eficienței arborilor 2-3-4 este mai dificilă decât analiza eficienței arborilor bicolori, dar echivalența dintre cele două categorii de arbori oferă un punct de pornire

Viteza

- Într-un arbore bicolor, fiecare operație de căutare conduce la vizitarea câte unui nod de pe fiecare nivel, indiferent dacă se caută un element existent sau locul de inserare pentru un nou element
- Numărul de niveluri dintr-un arbore bicolor (care este un arbore binar echilibrat) este aproximativ $\log_2(N+1)$, deci duratele de căutare vor fi proporționale cu această valoare

Viteza

- În arborii 2-3-4, trebuie să vizităm câte un singur nod de pe fiecare nivel, dar acești arbori au mai puține niveluri, în comparație cu arborii bicolori cu același număr de elemente

Observații

- În arborii 2-3-4, pot exista până la patru fii pentru fiecare nod
- Dacă fiecare nod este complet, înălțimea arborelui este direct proporțională cu $\log_4 N$
- Înălțimea unui arbore 2-3-4 reprezintă jumătate din cea a unui arbore bicolor cu același număr de noduri, dacă nodurile din arborele 2-3-4 sunt toate complete

Observații

- Din cauză că, în general, nu toate nodurile sunt complete, înălțimea unui arbore 2-3-4 variază între limitele $\log_2(N+1)/2$ și $\log_2(N+1)$
- Înălțimea redusă a arborilor 2-3-4 determină diminuarea corespunzătoare a timpului de căutare, în comparație cu arborii bicolori

Observații

- Fiecare nod conține mai multe elemente care trebuie examinate, ceea ce determină o creștere a timpului de căutare
- Din cauză că elementele dintr-un nod sunt examinate utilizând o căutare liniară, timpul de căutare va crește de M ori, M reprezentând numărul mediu de elemente din fiecare nod
- Timpul de căutare este proporțional cu $M \cdot \log_4 N$

Observații

- Unele noduri conțin un singur element, altele 2 sau 3
- Dacă estimăm că media numărului de elemente este 2, rezultă un timp de căutare proporțional cu $2 \cdot \log_4 N = \log_2 N$

Observații

- Pentru arborii 2-3-4, numărul mai mare de elemente din fiecare nod tinde să anuleze avantajul adus de înălțimea redusă a arborilor
- Timpii de căutare pentru un arbore 2-3-4 și pentru un arbore bicolor sunt aproximativ egali, fiind de ordinul $O(\log N)$

Memoria necesară

- Fiecare nod dintr-un arbore 2-3-4 necesită memorarea pointerilor către trei elemente și, respectiv, către cei patru fii
- Nu toată memoria alocată este utilizată efectiv

Memoria necesară

- Un nod cu un singur element va irosi două treimi din spațiul alocat pentru elemente și jumătate din cel alocat pentru fii
- Un nod cu două elemente va irosi o treime din spațiul alocat pentru elemente și un sfert din cel alocat pentru fii, deci va utiliza efectiv numai $5/7$ din spațiul alocat

Observații

- Presupunând că, în medie, nodurile au câte două elemente, rezultă că aproximativ $2/7$ din spațiul de memorare alocat este neutilizat
- Datorită echilibrării lor, arborii bicolori conțin mai puține noduri cu un singur fiu, deci aproape toată memoria alocată este utilizată efectiv

Observații

- Arborii bicolori sunt mai eficienți decât arborii 2-3-4, din punct de vedere al utilizării memoriei
- Din punct de vedere al programării, arborii 2-3-4 sunt mai ușor de implementat

Concluzii

- Arborii 2-3-4 sunt arbori multicăi, în care fiecare nod are cel mult trei elemente și cel mult patru fii
- În arborii 2-3-4, toate inserările se efectuează în nodurile frunză, care se află toate pe același nivel

Concluzii

- Arborii 2-3-4 conțin trei tipuri de noduri:
 - 2-noduri, care au o cheie și doi fii
 - 3-noduri, care au două chei și trei fii
 - 4-noduri, care au trei chei și patru fii
- Arborii 2-3-4 nu pot conține 1-noduri
- Căutarea într-un arbore 2-3-4 presupune examinarea cheilor din fiecare nod

Concluzii

- Dacă valoarea căutată nu se găsește în nodul curent, următorul nod ales în procesul de căutare va fi:
 - Fiul numerotat cu 0, când valoarea este mai mică decât cheia 0
 - Fiul 1, când valoarea este între cheia 0 și cheia 1
 - Fiul 2, când valoarea este între cheia 1 și cheia 2
 - Fiul 3, când valoarea este mai mare decât cheia 2

Concluzii

- Inserarea într-un arbore 2-3-4 presupune divizarea fiecărui nod complet, la parcurgerea descendentă a arborelui, în căutarea locului de inserare
- Divizarea rădăcinii creează două noduri noi în arbore, în timp ce divizarea oricărui alt nod creează un singur nod nou

Concluzii

- Înălțimea unui arbore 2-3-4 poate crește numai în urma divizării rădăcinii
- Înălțimea unui arbore 2-3-4 este cel mult egală cu $\log_2 N$
- Arborii 2-3-4 utilizează ineficient memoria, deoarece unele noduri nu sunt nici măcar pe jumătate complete