

Structuri de date – Curs 10

Conf. univ. dr. Cristian CIUREA
Departamentul de Informatica si Cibernetica Economica
Academia de Studii Economice din Bucuresti
cristian.ciurea@ie.ase.ro

Agenda

- ▶ Arbori multicăi de căutare
- ▶ Arbori B

Arbori B

- ▶ Sisteme de Gestiune a Bazelor de Date Relaționale (SGBDR): operație importantă este regăsirea rapidă a datelor pe baza de indecși.
- ▶ Indexul este o colecție de perechi $\langle \textit{valoare cheie}, \textit{adresa articol} \rangle$ pentru a facilita accesul la o colecție de articole.
- ▶ Structura de date foarte des folosită pentru implementarea indecșilor este arborele de căutare.

Arbori B

- ▶ Un index se numește *dens* dacă el conține câte o pereche $\langle \text{valoare cheie}, \text{adresa articol} \rangle$ pentru fiecare articol din colecție.
- ▶ Un index care nu este dens este numit index *rar*.

Arbori B

- ▶ Articolele memorate:
 - oricât de complexe;
 - conțin un câmp numit cheie pentru identificare.
- ▶ Dacă există o relație de ordine totală pe mulțimea cheilor posibile ce vor trebui regăsite cu ajutorul arborelui de căutare, atunci arborele de căutare se numește *bazat pe ordinea cheilor*.

Arbori B

- ▶ Arbori de căutare bazați pe ordinea cheilor:
 - *arbori binari de căutare*: o singură cheie asociată fiecărui nod;
 - *arbori multicăi de căutare*: mai multe chei asociate fiecărui nod.
- ▶ Performanțele unui index se îmbunătățesc prin mărirea factorului de ramificare a arborelui de căutare folosit.

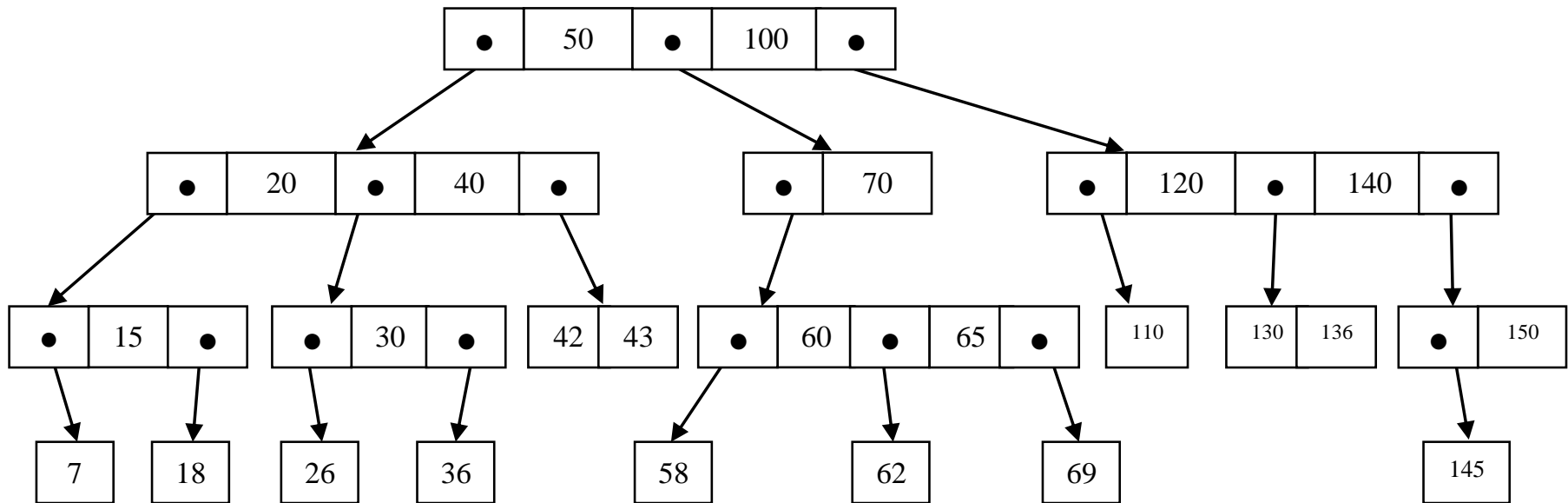
Arbori B

Arborii multicăi de căutare:

- ▶ generalizare a arborilor binari de căutare;
- ▶ nod oarecare: număr de M chei, ordonate strict crescător, ramificare în $M+1$ subarbori;
- ▶ M diferă de la nod la nod;
- ▶ M este între anumite limite pentru folosirea eficientă a mediului de stocare;
- ▶ M chei atașate unui nod formează o *pagină*.

Arbori B

- ▶ Arbore multicăi de căutare de ordin $n=3$:



Arbori B

Arbore multicăi de căutare – proprietăți:

- ▶ structura nodului:

n	P ₀	K ₀	P ₁	K ₁	P ₂	...	P _{n-1}	K _{n-1}	P _n
---	----------------	----------------	----------------	----------------	----------------	-----	------------------	------------------	----------------

- ▶ P_0, P_1, \dots, P_n pointeri către subarbori;
- ▶ K_0, K_1, \dots, K_{n-1} valorile cheilor.
- ▶ număr de ramificații – restricția $M \leq n - 1$.

Arbori B

- ▶ Arbore multicăi de căutare – proprietăți (continuare):
 - valorile cheilor într-un nod sunt în ordine crescătoare;
 - valorile cheilor din nodurile subarborelui P_i sunt mai mici decât valoarea cheii K_i , $i = 1, 2, \dots, n-1$;
 - valorile cheilor din nodurile subarborelui P_n sunt mai mari decât valoarea de cheie K_{n-1} ;
 - subarborii P_i sunt, de asemenea, arbori multicăi de căutare.

Arbori B

Arbore B de ordin N – proprietati:

- ▶ arbore multicăi de căutare;
- ▶ toate nodurile frunză sunt pe același nivel (arborele este echilibrat);
- ▶ rădăcina are cel puțin doi descendenți, dacă nu este frunză (ramificare timpurie);
- ▶ pagina conține cel puțin N chei; excepție rădăcina, care poate avea mai puține chei, dacă este frunză (un nod este cel puțin 50% plin);

Arbori B

- ▶ Arborii B sunt structuri de date dinamice arborescente utilizate pentru memorarea informațiilor necesare căutării rapide în baze de date foarte largi.

Arbori B

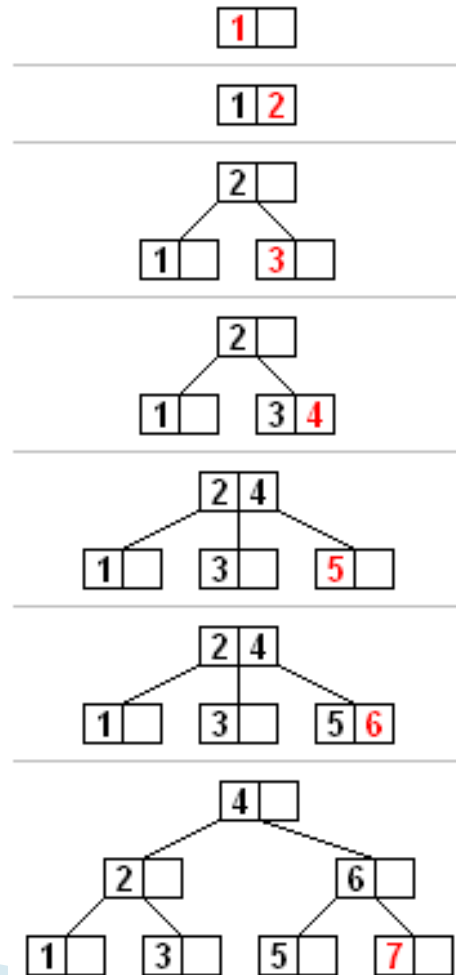
- ▶ Un arbore B de ordin N este o structură de date arborescentă în care nodurile și operațiile de inserare/ștergere respectă un set de reguli bine definite:
 - fiecare nod al arborelui poate conține la un moment dat maxim $2 \cdot N$ chei;
 - fiecare nod al arborelui conține minim N chei;
 - oricare nod al arborelui B de grad N nu conține mai mult de $2 \cdot N$ valori;
 - cheile din fiecare nod sunt sortate descrescător sau crescător;

Arbori B

- ▶ Arbori B – proprietăți (continuare):
 - operația de inserare/ștergere cheie are loc în noduri frunză;
 - toate nodurile frunză ale arborelui B de ordin N se găsesc pe același nivel;
 - înălțimea unui arbore B de grad N ce conține n chei este egală cu $\log_N n$.

Arbori B

- ▶ Arbore B de ordinul $N=7$:



Arbori B

- ▶ Arborele B este o structura de date de tip N-ară. Această caracteristică este dată de faptul ca în arborele B de ordin N nodurile părinte au maxim $2*N+1$ noduri fii.
- ▶ Arborele B este utilizat pentru a realiza operații de căutare în baze de date stocate pe suporturi externe, datorită dimensiunii lor foarte mari.
- ▶ Principiul de la care s-a pornit în dezvoltarea arborilor B a fost minimizarea timpului de acces la disk, în momentul căutării unei chei.

Arbori B

Operații de bază:

► Căutarea:

- Comparație cheie căutată cu cheile nodului curent;
- Nodul de start: rădăcina
- Lungimea maximă a drumului de căutare este dată de înălțimea arborelui
- Situații de continuare a cautarii:
 - $c_i < x < c(i + 1)$: căutare în nodul indicat de P_i ;
 - $c_n < x$: căutare în nodul indicat de P_n ;
 - $x < c_0$: căutare în nodul indicat de P_0 .

Arbori B

Operații de bază (continuare):

► Inserarea unei chei în arbore B:

- precedată de operația de căutare;
- cheie găsită în arbore: abandon operație inserare;
- cheia nu a fost găsită: căutare terminată într-un nod frunză, unde se inserează noua cheie;

► Situații de inserare:

- nodul are mai puțin de 2^*N chei: inserare fără modificarea structurii arborelui B;
- nodul are numărul maxim de 2^*N chei: “fisionare” nod; rezultă două noduri care se vor găsi pe același nivel și o cheie mediană care nu se va mai găsi în nici unul din cele două noduri.

Arbori B

- ▶ Operația de inserare a unei noi valori într-un arbore B are în centru un nod frunză, iar după orice inserare arborele este tot unul B.

Arbori B

- ▶ Inserarea unei noi valori, *CheieNoua*, într-un arbore B de ordin N existent necesită parcurgerea pașilor:
 - căutarea nodului în care se face inserarea;
 - poziționarea pe un nod frunză, care nu are noduri fiu, determină inserarea noii valori, *CheieNoua*, în șirul sortat crescător de chei fără a deteriora ordonarea;
 - verificarea numărului de chei din acest nod pentru a vedea dacă cu această nouă valoare nu a avut loc depășirea numărului maxim de valori, $2*N$;
 - dacă nodul conține $2*N+1$ chei atunci arborele trebuie reechilibrat;

Arbori B

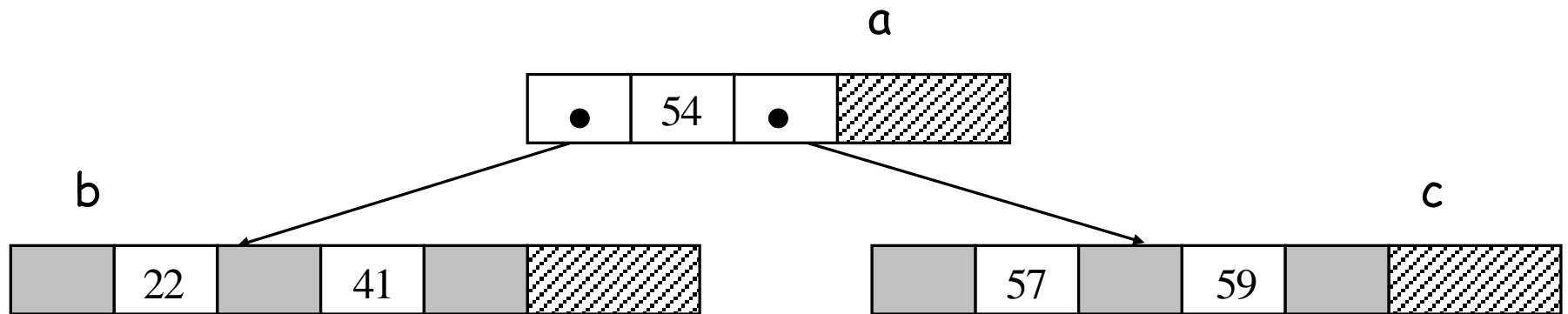
Arbore B de ordin 2:

- ▶ Numărul maxim de chei dintr-un nod este 4;
- ▶ Inserarea valorilor de cheie 22, 57, 41, 59.

	22		41		57		59	
--	----	--	----	--	----	--	----	--

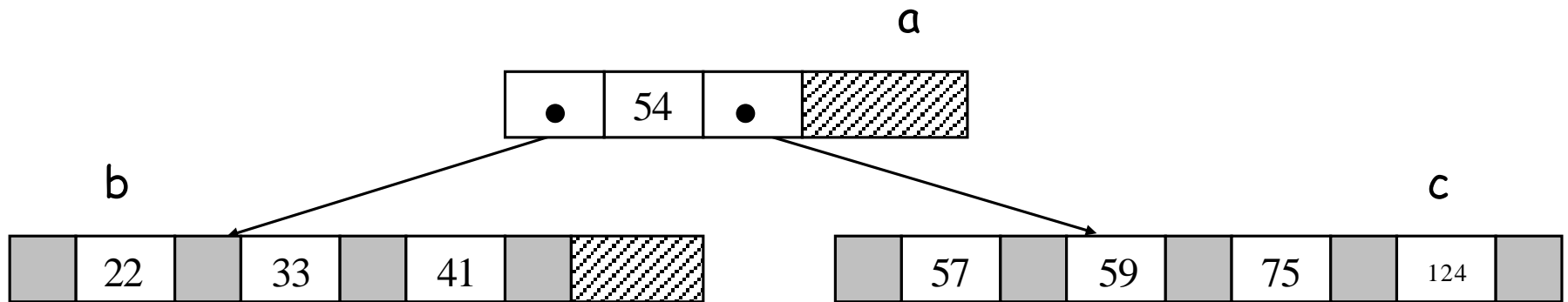
Arbori B

- ▶ În urma inserării cheii 54, nodul rădăcină va conține prea multe chei, așa că el va “fisiona”:



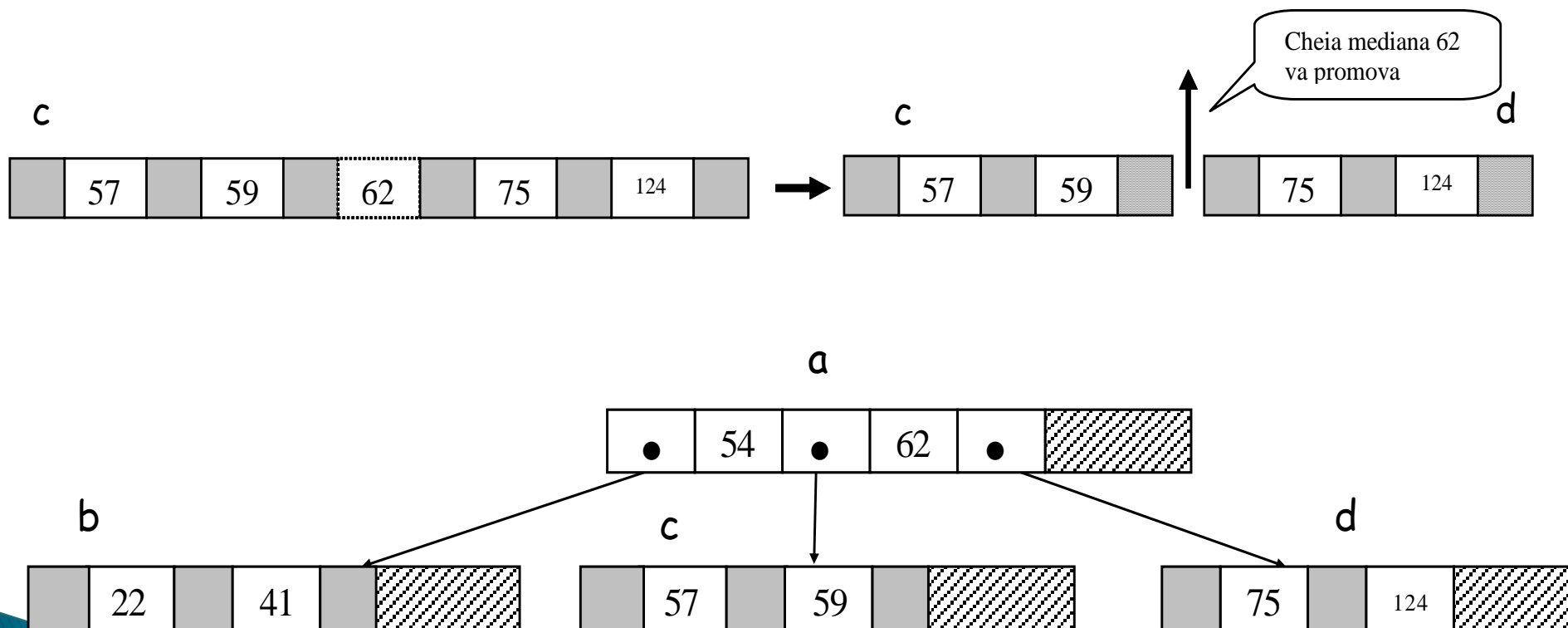
Arbori B

- Inserarea cheilor 33, 75, 124:



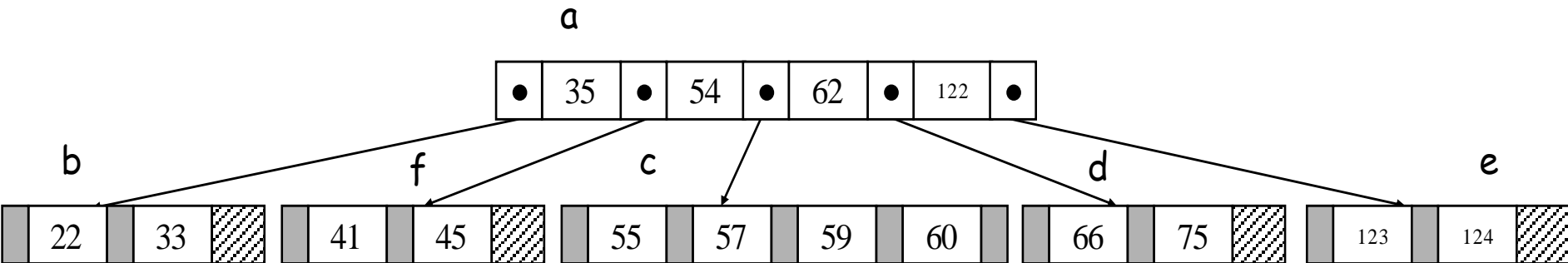
Arbori B

- ▶ Inserarea cheii 62: divizarea nodului *c*.
- ▶ Cheia 62 promovează în nodul rădăcină:



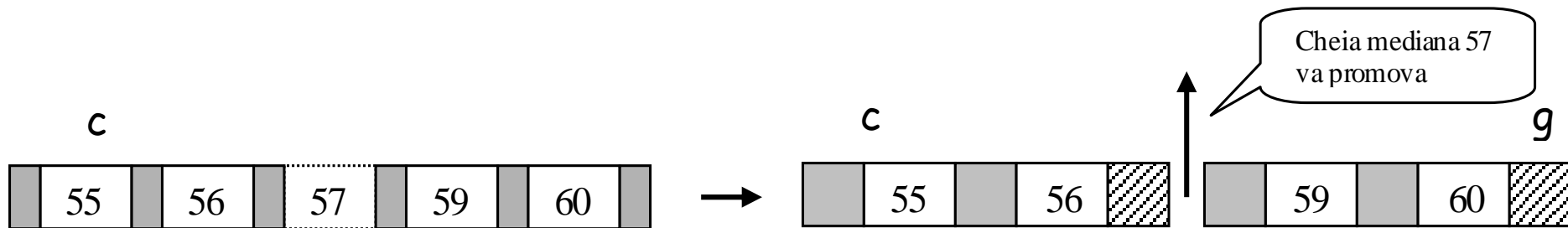
Arbori B

- Inserarea cheilor 33, 122, 123, 55, 60, 45, 66, 35:



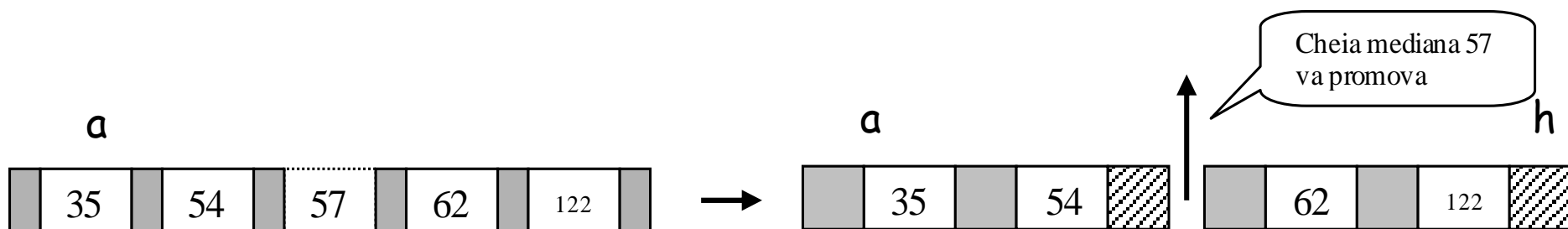
Arbori B

- ▶ Inserarea cheii 56: fisionarea nodului c .



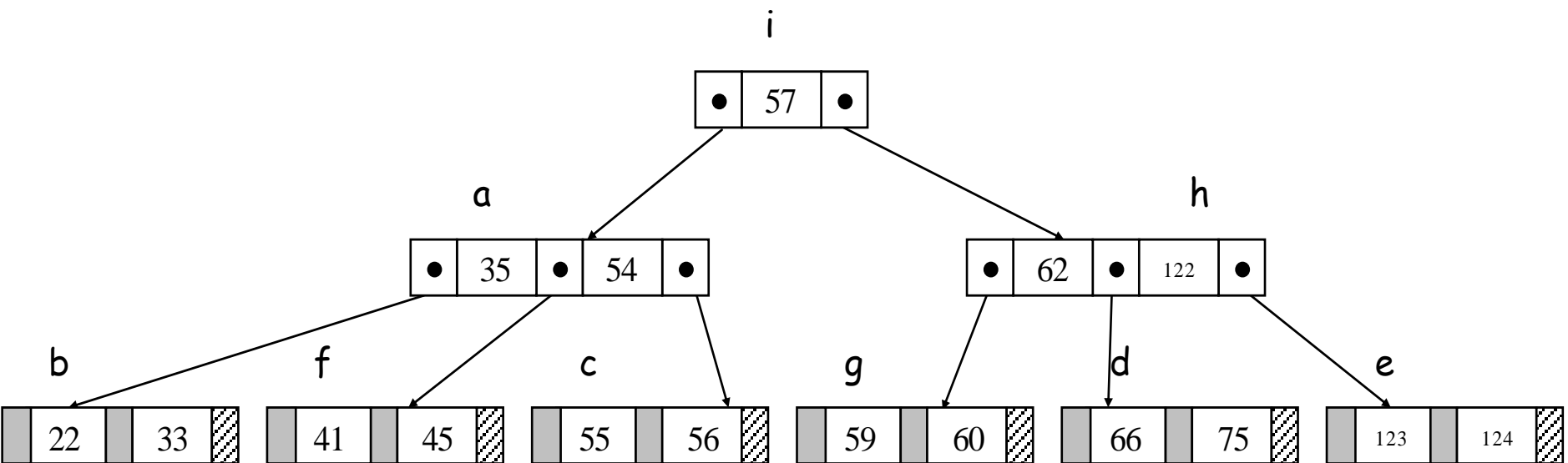
Arbori B

- ▶ Nodul părinte a este plin și nu poate primi noua cheie 57. Algoritmul de fisionare este aplicat din nou, pentru nodul a .



Arbori B

- ▶ Noua structură a arborelui B:



Arbori B

- ▶ Algoritmul de inserare a unei chei într-un arbore B:
 - inserează noua valoare de cheie în nodul frunză corespunzător;
 - `nodul_curent = nodul_frunză`;
 - `while (nodul_curent este OVERFLOW)`:
 - divide `nodul_curent` în două noduri aflate pe același nivel și promovează cheia mediană în nodul părinte pentru `nodul_curent`;
 - `nodul_curent = nodul_părinte` pentru `nodul_curent`.

Arbori B

- ▶ Cel mai rău caz: aplicarea algoritmului de fisionare pe întreaga înălțime a arborelui, fisionându-se $h-1$ noduri (h este înălțimea arborelui înainte de inserare).

Arbori B

Operații de bază (continuare):

- ▶ Ștergerea unei chei dintr-un arbore B:
 - simplă, dacă valoarea de cheie ștearsă se află într-un nod frunză;
 - complexă, dacă valoarea de cheie ștearsă nu se află într-un nod frunză: ștergere logică, fiind înlocuită cu o altă cheie, vecină în inordine, care va fi ștearsă fizic.

Arbori B

- ▶ Dacă în cazul operației de inserare, nodul arborelui nu trebuie să dețină mai mult de $2*N$ chei, în cazul ștergerii se urmărește ca nodul din care s-a șters să nu rămână cu mai puțin de N chei.

Arbori B

- ▶ Ștergerea valorii *ValCheie* dintr-un arbore B de ordin N implică parcurgerea pașilor:
 - căutarea nodului ce conține cheia cu valoarea de șters;
 - dacă valoarea de șters se află în nodul rădăcină și acesta este unicul nod al arborelui, adică nu are noduri fiu, atunci valoarea este ștearsă;
 - dacă valoarea *ValCheie* se găsește într-un nod frunză, iar acesta conține mai mult de N chei, atunci ștergerea are loc efectiv.

Arbori B

- ▶ Pași operație ștergere (continuare):
 - dacă valoarea *Val/Cheie* se găsește într-un nod rădăcină ce are noduri fiu, atunci, fie se alege cea mai mare valoare din nodul fiu stânga, cu care se înlocuiește valoarea ștearsă, fie se alege cea mai mică valoare din nodul fiu dreapta;
 - dacă valoarea de șters se găsește într-un nod frunză ce are exact N chei, procesul de eliminare este urmat de un împrumut de la unul dintre nodurile vecine, stânga sau dreapta, ce conțin mai mult de N chei;

Arbori B

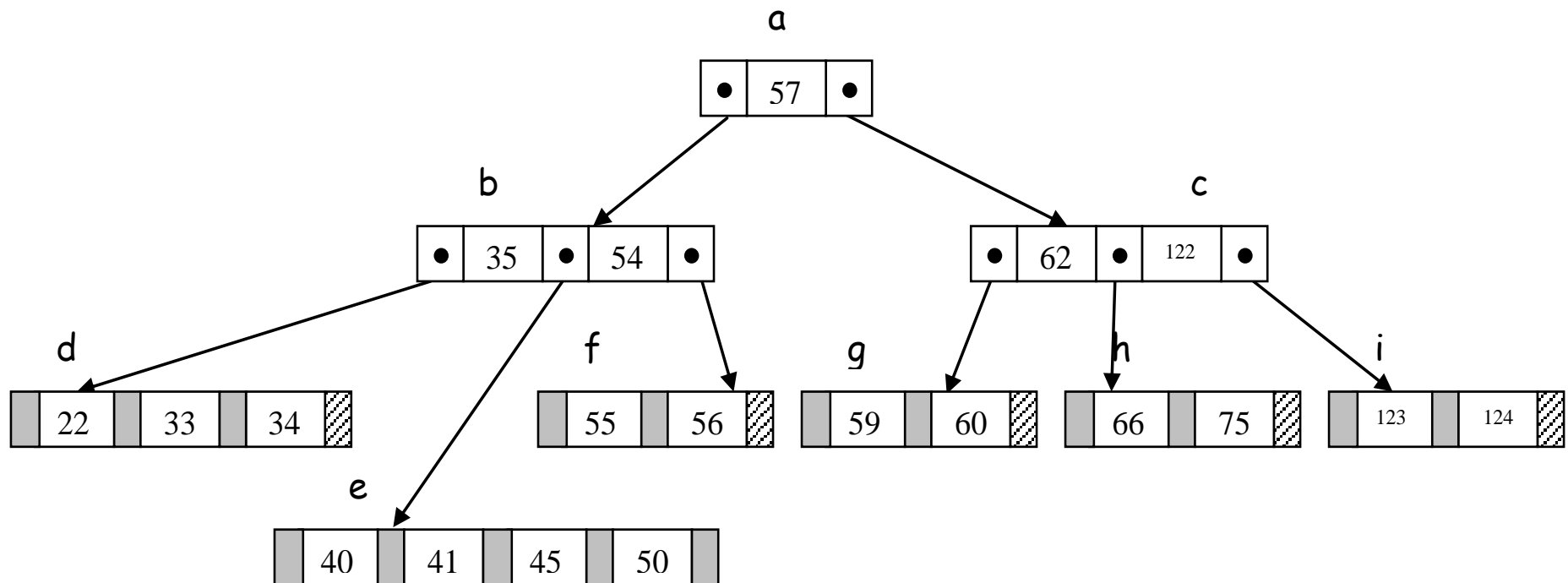
- ▶ Situații de ștergere:
 - nodul conține mai mult de N chei: ștergerea nu ridică probleme;
 - nodul are numărul minim de chei N : după ștergere, numărul de chei din nod va fi insuficient; se împrumută o cheie din nodul vecin cu cel puțin N chei (partajare).

Arbori B

- ▶ Ștergerea unei chei dintr-un arbore B (continuare):
 - Dacă nu se poate face o partajare (nodurile vecine au numărul minim de chei), atunci două noduri vecine vor fuziona, împrumutându-se o cheie și din nodul părinte.
 - Partajarea sau fuzionarea trebuie eventual repetate și pentru nivelurile superioare.
 - Cazul cel mai nefavorabil: partajarea sau fuzionarea parcurg întreaga înălțime a arborelui; se va forma o nouă rădăcină, înălțimea arborelui scăzând cu un nivel.

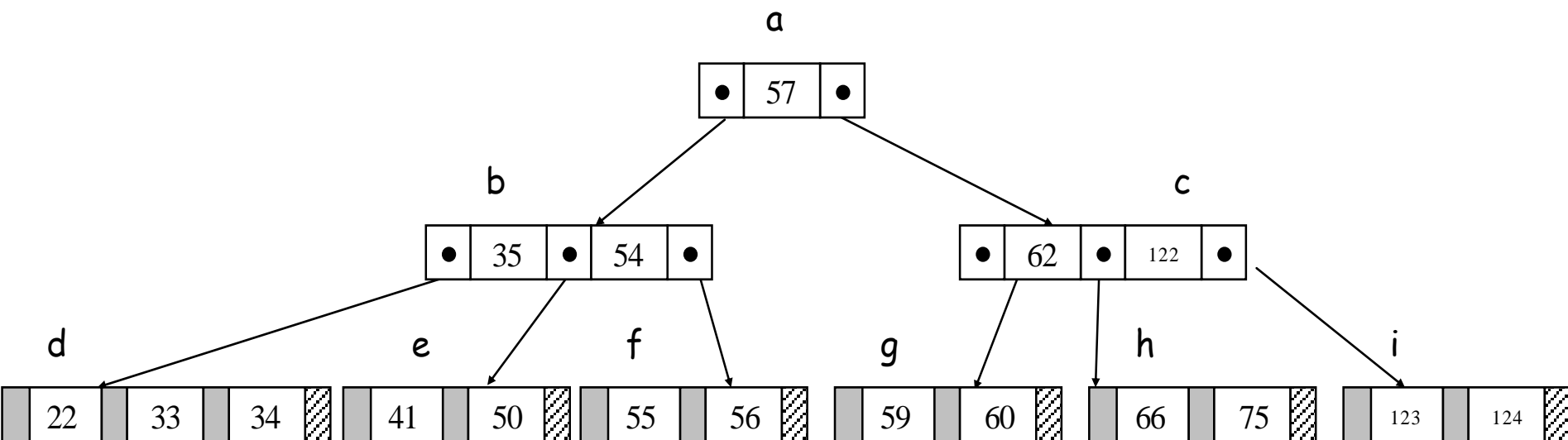
Arbori B

- Se consideră următoarea configurație de arbore B de ordin 2:



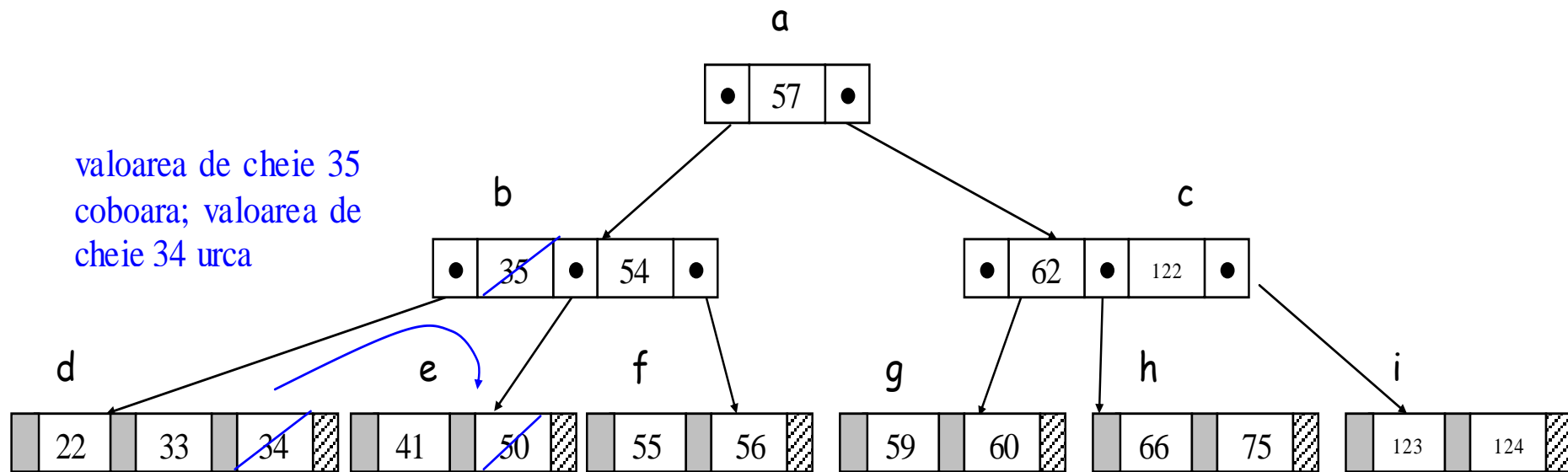
Arbori B

- ▶ Ștergerea valorilor de cheie 40 și 45 din nodul *e* nu ridică probleme:



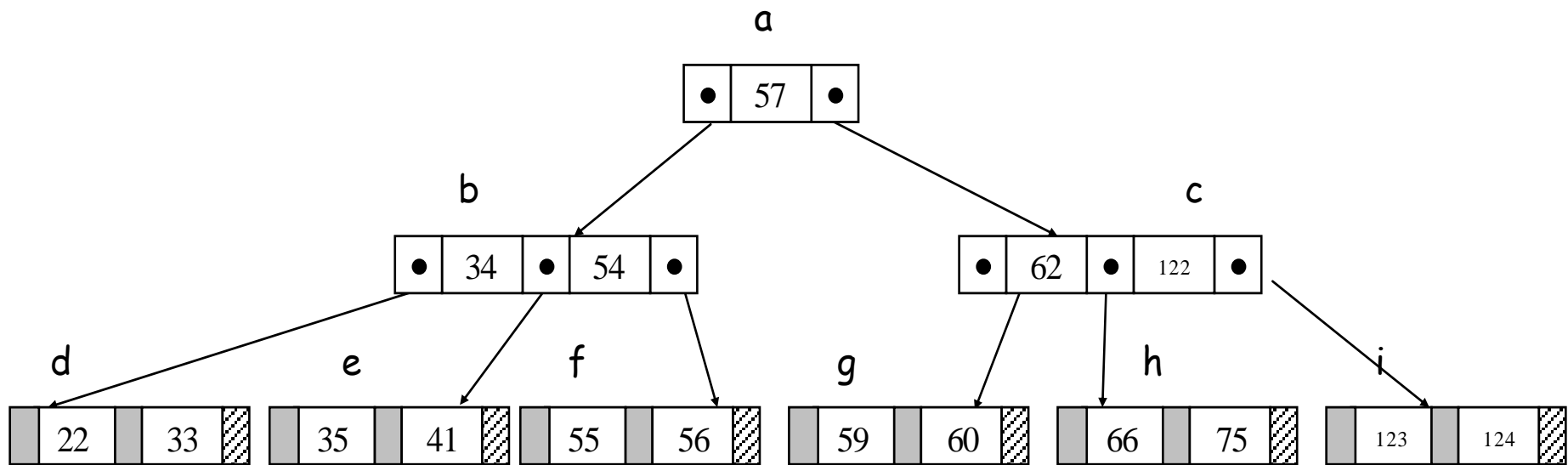
Arbori B

- ▶ Ștergerea valorii de cheie 50: partajare între nodurile d și e :



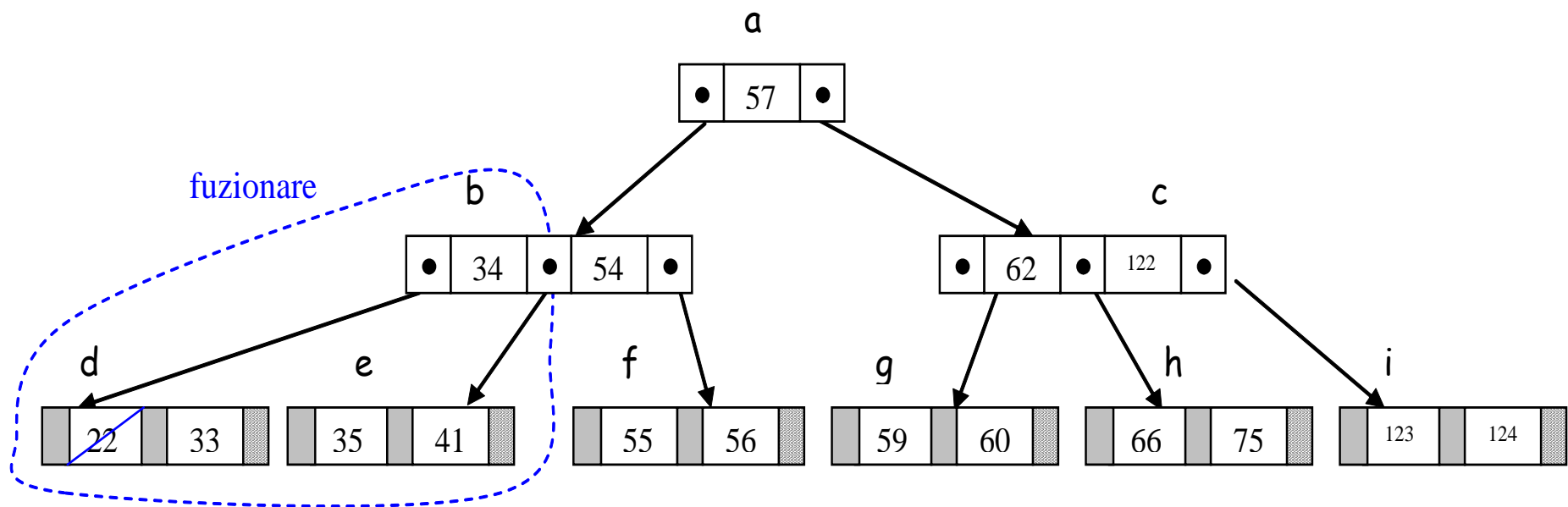
Arbori B

- ▶ Structura arborelui după partajare:



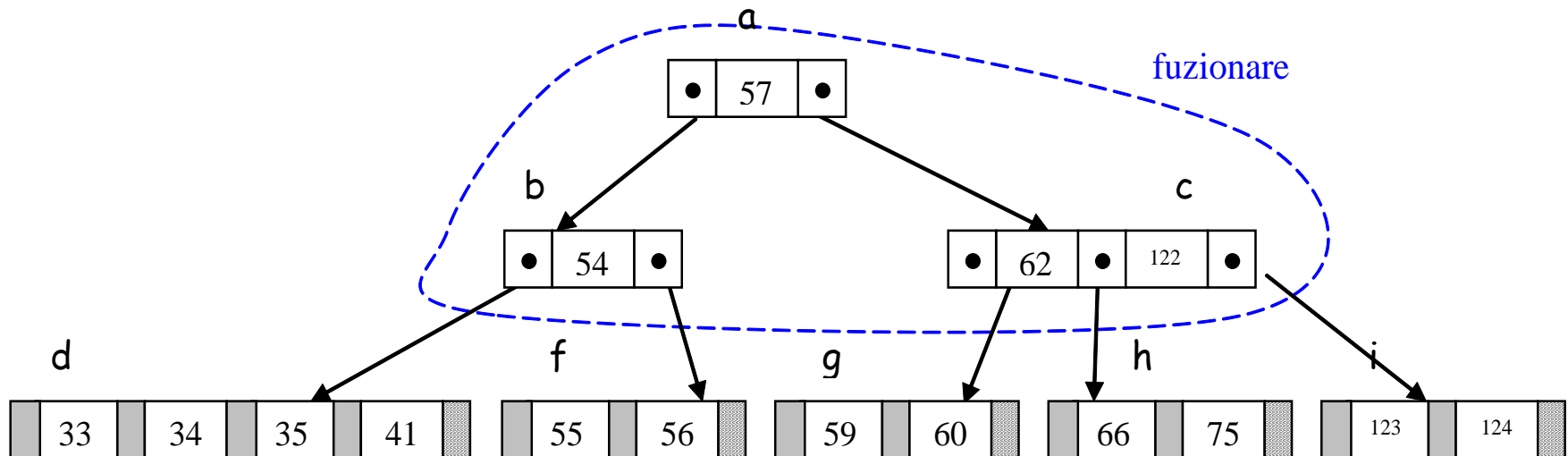
Arbori B

- ▶ Ștergerea valorii de cheie 22: fuzionarea nodurilor d și e :



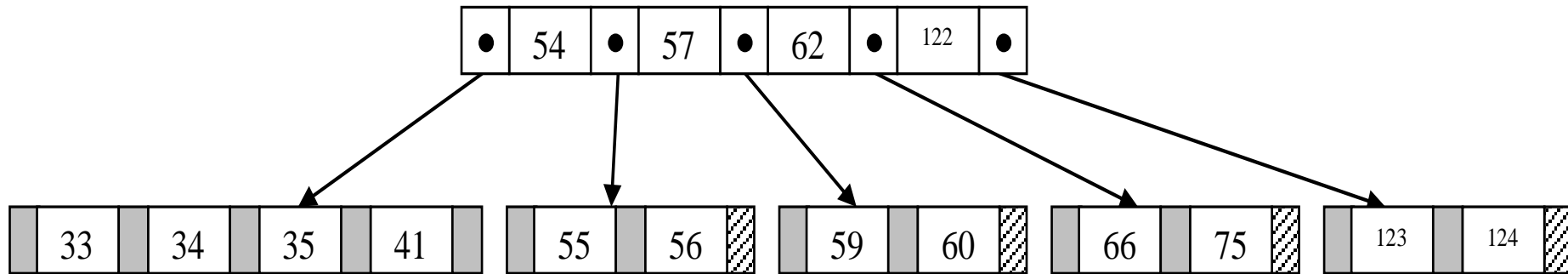
Arbori B

- ▶ În urma fuzionării nodurilor d și e , nodul b va conține prea puține valori de cheie: fuzionare nodurile b și c .



Arbori B

- ▶ Structura finală a arborelui B:



Arbori B

Algoritmul de ștergere dintr-un arbore B:

- ▶ dacă valoarea cheii care se șterge nu este într-un nod frunză, atunci se înlocuiește valoarea cheii cu cheia succesori/predecesor;
- ▶ `nodul_curent = nodul_frunza`;
- ▶ `while (nodul_curent este UNDERFLOW)`:
 - încearcă partajarea cu unul din nodurile vecine aflate pe același nivel, via nodul părinte;
 - dacă nu este posibilă partajarea, atunci:
 - 1. fuzionează `nodul_curent` cu un nod vecin, folosind o valoare de cheie din nodul părinte;
 - 2. `nodul_curent = nod_părinte` pentru `nodul_curent`.

Bibliografie

- ▶ Ion Ivan, Marius Popa, Paul Pocatilu (coordonatori) – *Structuri de date*, Editura ASE, București, 2008.
 - Cap. 13. Arbori B