

# Arbori binari de căutare





## Căutare

Găsirea unei anumite informații sau părți de informație dintr-un volum mare de date memorate/stocate anterior este o operație fundamentală, numită *căutare*, a majorității aplicațiilor pe calculator.

Datele sunt organizate ca **articole** sau item-uri fiecare având o **cheie** care se utilizează în căutare. Scopul căutării este găsirea articolelor care au chei egale cu cheia de căutare.

**Scopul** căutării este accesarea informației din articol în vederea prelucrării.



## Tabelă de simboluri

**Definiție:** O **tabelă de simboluri** este o structură de date care suportă două operații de bază: **inserează** un nou articol și **returnează** un articol cu o cheie dată.

Tabelele de simboluri sunt numite și **dicționare** prin analogie cu sistemul secular de a da definiții ale cuvintelor listându-le alfabetic într-o carte de referință:

- *cheile* sunt cuvintele
- *articolele* sunt înregistrările asociate cu cuvintele și conțin definiția, pronunția, etimologia, etc



**căblu, căbluri**, substantiv Conductă formată din unul sau mai multe fire metalice, învelite într-un material izolator. *Înainte de a umbra la căblu, vezi să nu fie conectat la curent.*

**cacáo**, substantiv Praf obținut din semințele măcinate ale arborelui de cacao din care se prepară ciocolata, unele băuturi. *Lapte cu cacao.*

**căctus, căctuși**, substantiv Plantă din țările calde, cu tulpină cămoasă și acoperită cu spini, care înflorește rar. *Căctusul este cultivat la noi ca plantă decorativă.*



**cădă, căzi**, substantiv Vas mare în care se face baie. *Cadă de metal.*



**cadóu, cadóuri**, substantiv Obiect primit de la cineva sau oferit cuiva la anumite sărbători. I se mai spune **dar**. *Am primit o mulțime de cadouri la ziua de naștere*

**cafeá, caféle**, substantiv Praf obținut din semințele prăjite și măcinate ale arborelui de cafea. Se folosește în alimentație. © Tot **cafeaua** se spune băuturi preparate din acest praf. *Cafeaua este o băutura care te înviorează.*

**caiet, caiete**, substantiv Foi de hârtie legate împreună și folosite la scris sau la desenat. *Treceți exercițiul în caiet.*

**cais, caiși**, substantiv Pom fructifer cu flori roz și fructe rotunde, dulci și cămoase, de culoare portocalie. *Caisul înflorește primăvara devreme.*

**caisă, caise**, substantiv Fructul caisului. *Dulceață de caise.*

**- B -**

**Backup**

Operatie de copiere a unor informatii mai importante de pe hard discul local pe un drive extern, pentru a asigura existenta unei copii de siguranta in cazul in care datele de pe hard disc ar putea fi avariate. Backup-urile se realizeaza in principal pe benzi magnetice sau pe dischete.

**Banda magnetica**

Dispozitiv extern de stocare a informatiilor electronice. Mediul de stocare al informatiilor este o banda din material plastic, acoperita cu un strat de substante chimice ce pot fi usor magnetizate (asemanatoare benzilor din casetele audio). Inscriptiunea informatiilor pe aceste benzi magnetice se realizeaza prin magnetizarea stratului de substante chimice cu ajutorul unui "cap de scriere". benzile magnetice sunt foarte des utilizate in realizarea backup-urilor.

**Baud**

Unitatea de masura pentru largimea de banda a unei legaturi la Internet. Reprezinta numarul maxim de semnale ce pot fi trimise într-o secunda, fiind de asemenea referita sub numele de rata de semnalizare. BAUD-ul este uzual asociat cu viteza de operare a unui modem. Un Baud reprezinta aproximativ un bit transmis pe secunda

**Baza de date**

Colectie de date si informatii organizate intr-un fisier. Bazele de date se compun din mai multe inregistrari, iar aceste inregistrari sunt alcatuite din mai multe campuri. Datele sau informatiile efective sunt stocate in aceste campuri. Exista software comercial specializat pentru lucrul cu bazele de date, procesarea informatiilor si intretinerea acestora.

**BBS**

(Bulletin Board System) Un serviciu mai vechi si mai putin utilizat în prezent. BBS-urile reprezinta în general calculatoarele de pe care utilizatorii conectati pot sa realizeze download-uri de software. BBS-urile sunt utilizate aproape exclusiv cu rolul de bibliotecii de programe shareware si freeware, dar si pentru a lasa mesaje publice

**BCNU**

(Jargon Chat) "Ne mai vedem noi" (Be seeing you)

**(Format) Binar (de fisier)**

Un fisier ce contine informatie in format binar contine, pe langa text (respectiv caractere direct afisabile) si alte caractere de control. De exemplu, un fisier ce contine un text realizat cu ajutorul unui procesor de texte, pe langa textul propriu-zis, mai contine si caractere ce specifica formatarea fiecarei portiune de text. De asemenea, un exemplutpic de fisiere cu date stocate in format binar sunt arhivele si fisierele direct executabile.

**BMP**

(Extensie de fisier) Marcheaza un fisier ce contine o informatie grafica (poza, desen...). BMP este unul dintre cele mai vechi formate de stocare a informatiei. Dimensiunea unui fisier de acest tip lasa însa foarte mult de dorit, acest inconvenient fiind solutionat prin elaborarea altor formate grafice (vezi JPG si GIF)

**BRB**

(Jargon Chat) Ma întorc imediat (Be right back)

**Browser**

Program utilizat pentru a accesa informatia disponibila pe Internet, organizata sub forma de pagini web. Browserele va permit sa navigati pe Internet si sa utilizati majoritatea serviciilor de Internet disponibile. Cele mai cunoscute browsere sunt Netscape Navigator si Microsoft Internet Explorer, acestea fiind disponibile gratuit

**BTS**

(Jargon Chat) Bust/Talie/Sold.

**BTW**

(Jargon Chat) Apropo (By The Way)

**Eroare de programare (Bug)**

Acest termen se refera la o eroare aparuta într-un program, care la o prima vedere poate trece neobservata, însa poate determina respectivul program sa aiba în anumite situatii comportari dintre cele mai ciudate. Uneori poate fi vorba de un simplu semn gresit pus într-o relatie matematica din cadrul programului. În general companiile producatoare de software isi angajeaza persoane care sa le testeze programele înainte de a le lansa pe piata pentru a descoperi la timp aceste erori





## Avantajele tabelor de simboluri pe calculator :

- dispun de algoritmi eficienți de căutare,
- operații eficiente de inserare,
- operații eficiente de ștergere sau modificare,
- operații de combinare a 2 tabele într-una singură.

Indispensabile în organizarea software-ului pe calculator: cheile sunt numele simbolice, iar articolele conțin informații care descriu obiectul numit.

## TAD Tabelă de simboluri

<i>Insert /Inserează</i>	un nou articol.
<i>Search /Caută</i>	un articol /articole având o cheie dată.
<i>Delete /Șterge</i>	un articol specificat.
<i>Select</i>	al k-lea articol într-o tabelă de simboluri.
<i>Sort</i>	tabela de simboluri (vizitează toate articolele în ordinea cheilor)
<i>Join</i>	două tabele de simboluri
<i>Plus adeseori</i>	
<i>initialize</i>	
<i>test ifempty</i>	
<i>destroy</i>	
<i>copy</i>	



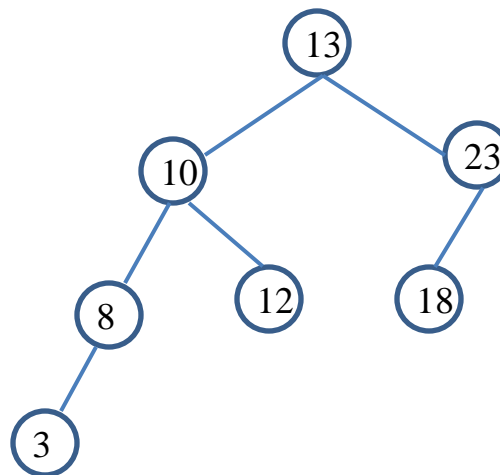
## Interfață TAD Tabelă de simboluri

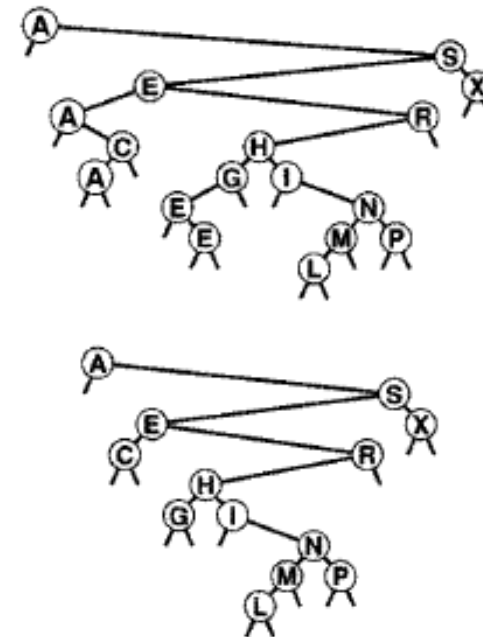
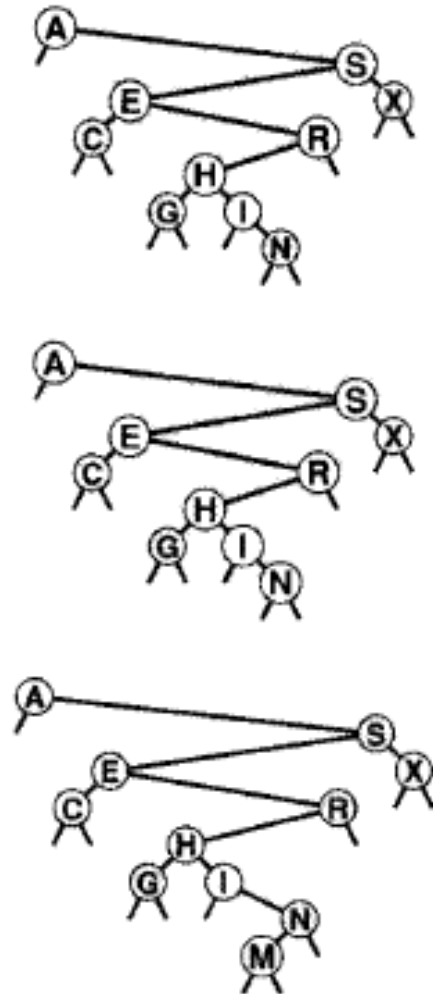
```
void STinit(int);  
int STcount ();  
void STinsert(Item);  
Item STsearch(Key);  
void STdelete(Item);  
Item STselect(int);  
void STsort(void (*visit) (Item));
```

---

## Arbori binari de căutare

**Definiție:** Un arbore binar de căutare (BST – Binary Search Tree) este un arbore binar care are asociată o cheie cu fiecare din nodurile sale interne, cu proprietatea suplimentară că cheia fiecărui nod este mai mare sau egală cu cheile din toate nodurile subarborelui său stâng și mai mică sau egală cu cheile din toate nodurile subarborelui său drept.






**Figure 12.7**  
Duplicate keys in BSTs

When a BST has records with duplicate keys (top), they appear scattered throughout the tree, as illustrated by the three highlighted A's. Duplicate keys do all appear on the search path for the key from the root to an external node, so they can readily be accessed. However, to avoid confusing usages such as "the A below the C," we use distinct keys in our examples (bottom).

(Sedgewick)

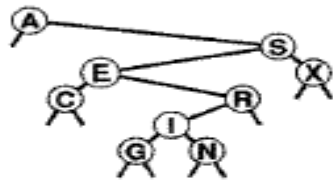
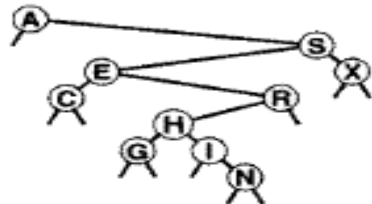
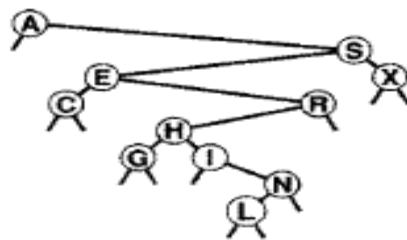
## Tabelă de simboluri implementată cu arbore binar de căutare

```
#include <stdlib.h>
#include "Item.h"
typedef struct STnode* link;
struct STnode { Item item; link l, r; int N; };
static link head, z;
link NEW(Item item, link l, link r, int N)
{ link x = malloc(sizeof *x);
  x->item = item; x->l = l; x->r = r; x->N = N;
  return x;
}
void STinit()
{ head = (z = NEW(NULLitem, 0, 0, 0)); }
int STcount() { return head->N; }
Item searchR(link h, Key v) //cautare in BST
{ Key t = key(h->item);
  if (h == z) return NULLitem;
  if eq(v, t) return h->item;
  if less(v, t) return searchR(h->l, v);
  else return searchR(h->r, v);
}
Item STsearch(Key v)
{ return searchR(head, v); }
link insertR(link h, Item item) //inserare in BST
{ Key v = key(item), t = key(h->item);
  if (h == z) return NEW(item, z, z, 1);
  if less(v, t)
    h->l = insertR(h->l, item);
  else h->r = insertR(h->r, item);
  (h->N)++; return h;
}
void STinsert(Item item)
{ head = insertR(head, item); }
```



Definim **succesorul** unui nod  $x$ , nodul  $y$  cu cea mai mică valoarea a cheii dar cu  $cheie[y] \geq cheie[x]$

Definim **predecesorul** unui nod  $x$ , nodul  $y$  cu cea mai mare valoarea a cheii dar cu  $cheie[y] \leq cheie[x]$



**Figure 12.19**  
**BST node deletion**

*This sequence depicts the result of deleting the nodes with keys L, H, and E from the BST at the top. First, the L is simply removed, since it is at the bottom. Second, the H is replaced with its right child, the I, since the left child of I is empty. Finally, the E is replaced with its successor in the tree, the G.*

### Program 12.15 Deletion of a node with a given key in a BST

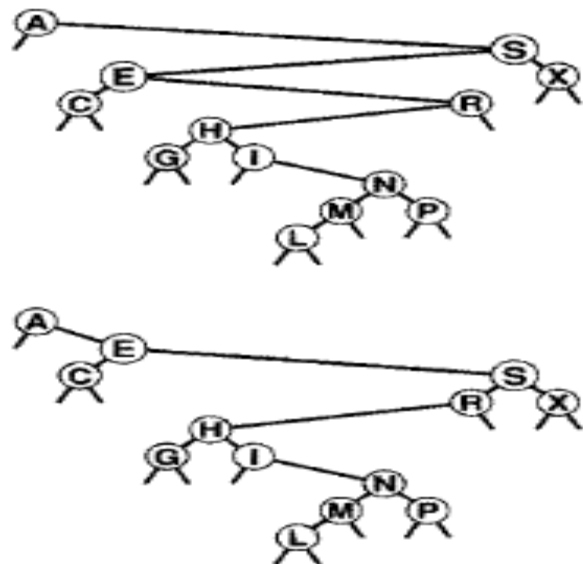
This implementation of the *delete* operation removes the first node with key *v* encountered in the BST. Working top down, it makes recursive calls for the appropriate subtree until the node to be deleted is at the root. Then, it replaces the node with the result of combining its two subtrees—the smallest node in the right subtree becomes the root, then its left link is set to point to the left subtree.

```
link joinLR(link a, link b)
{
    if (b == z) return a;
    b = partR(b, 0); b->l = a;
    return b;
}

link deleteR(link h, Key v)
{ link x; Key t = key(h->item);
  if (h == z) return z;
  if (less(v, t)) h->l = deleteR(h->l, v);
  if (less(t, v)) h->r = deleteR(h->r, v);
  if (eq(v, t))
    { x = h; h = joinLR(h->l, h->r); free(x); }
  return h;
}

void STdelete(Key v)
{ head = deleteR(head, v); }
```

(Sedgewick)

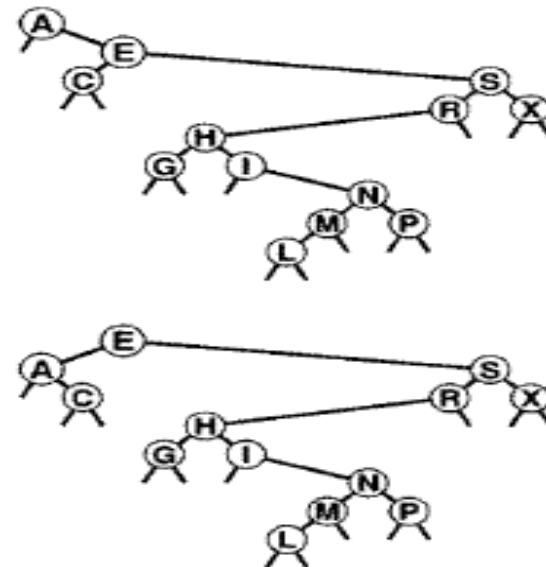


**Figure 12.12**  
**Right rotation in a BST**

*This diagram shows the result (bottom) of a right rotation at S in an example BST (top). The node containing S moves down in the tree, becoming the right child of its former left child.*

*We accomplish the rotation by getting the link to the new root E from the left link of S, setting the left link of S by copying the right link of E, setting the right link of E to S, and setting the link to S from A to point to E instead.*

*The effect of the rotation is to move E and its left subtree up one level, and to move S and its right subtree down one level. The rest of the tree is not affected at all.*



**Figure 12.13**  
**Left rotation in a BST**

*This diagram shows the result (bottom) of a left rotation at A in an example BST (top). The node containing A moves down in the tree, becoming the left child of its former right child.*

*We accomplish the rotation by getting the link to the new root E from the right link of A, setting the right link of A by copying the left link of E, setting the left link of E to A, and setting the link to A (the head link of the tree) to point to E instead.*

(Sedgewick)





Aceste două rutine efectuează operația de rotire într-un arbore BST

O **rotație la dreapta** face vechea rădăcină arborele drept al noii rădăcini (vechiul subarbore stâng al rădăcinii); **rotația la stânga** face vechea rădăcină subarborele stâng al noii rădăcini

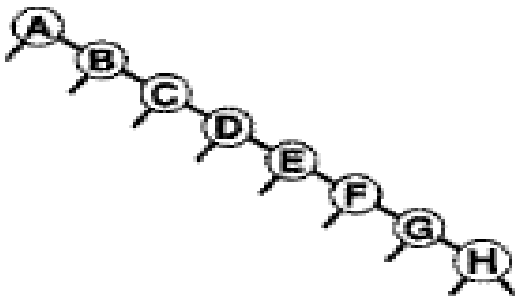
```
link rotR(link h)
{ link x = h->l; h->l = x->r; x->r = h;
  return x; }
link rotL(link h)
{ link x = h->r; h->r = x->l; x->l = h;
  return x; }
```

---

**Proprietatea1.** Căutarea necesită în medie aproximativ  $2\ln N \sim 1,39\ln N$  comparații într-un arbore binar de căutare format cu  $N$  chei aleatoare.

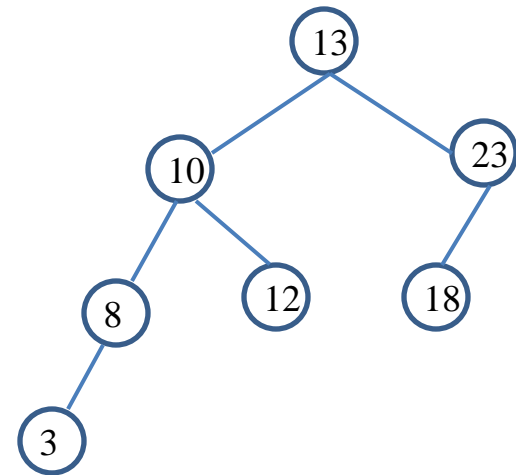
**Proprietatea2:** Inserarea și căutarea fără succes necesită în medie aproximativ  $2\ln N \sim 1,39\ln N$  comparații într-un arbore binar de căutare format cu  $N$  chei aleatoare.

**Proprietatea3:** În cel mai defavorabil caz, o căutare într-un arbore binar de căutare cu  $N$  chei necesită  $N$  comparații.



## Sortare cu arbore binar de căutare

```
//prin traversare inordine a BST  
  
void sortR(link h, void (*visit)(Item))  
{  
    if (h == z) return;  
    sortR(h->l, visit);  
    visit (h->item) ;  
    sortR(h->r, visit);  
}  
  
void STsort(void (*visit) (Item))  
{ sortR(head, visit); }
```



3 8 10 12 13 18 23

[citatepedia.info](http://citatepedia.info)

Cauta si pastreaza mereu o  
bucata de cer deasupra vietii  
tale.

Marcel Proust

