Алгоритми и структури на податоци

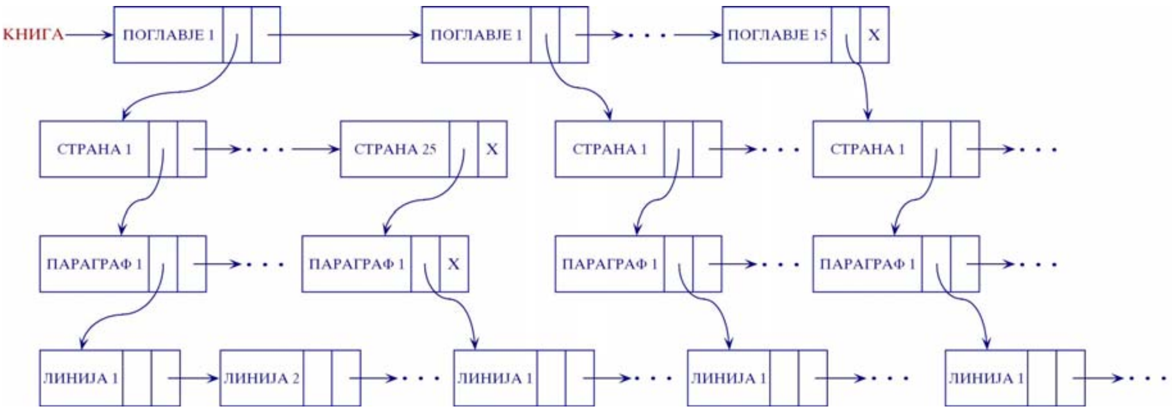
– белешки од предавања –

5. Стебла

5.1 Комплексни листи

Динамичките структури што беа прикажани во предходните поглавја описно можеа да се опишат со аналогијата на синџири. Имено имавме информација која беше поврзана со друга информација во секвентен распоред. Иако информацијата можеше да биде било каков запис, поради нивната секвентност (последователност) обично се вели дека таквите податочни структури се едноставни динамички податочни структури.

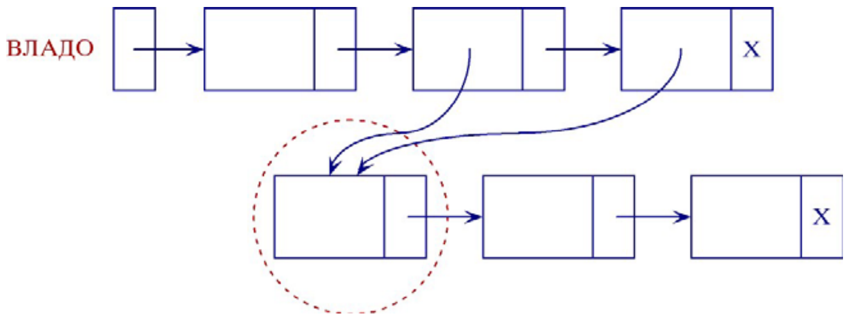
Сложените динамички податочни структури може да се опишат како комплексни листи. Тоа се листи каде еден јазел може да покажува кон нова листа (така наречена подлиста). Постојат голем број примери од реалниот свет кои укажуваат на потреба од воведување на вакви податочни структури. Еден таков пример е потребата од опис на еден текст на книга (при имплементација на текст едитор). Една можна претстава на книгата е дадена со податочната структура од слика 5.1. Книгата се состои од повеќе поглавја, кои можат да бидат на повеќе страни, кои содржат параграфи составени од линии. Текст едиторот во главно треба да овозможи манипулација (внесување или бришење) на јазлите (на пример параграфи), или пребарување по содржината на некој од јазлите.



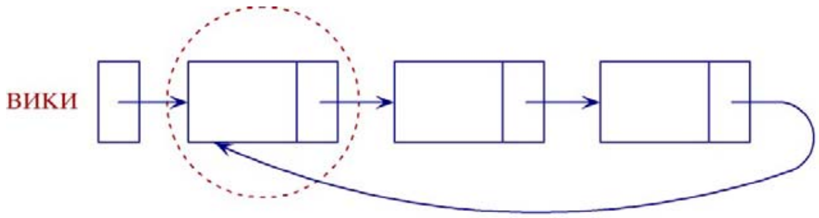
слика 5.1 Претставување на една книга како комплексна листа

Од примерот се гледа дека кон еден јазел може да се придружи комплетно нова листа. Иако се добива релативно комплексна структура, треба да се укаже на фактот дека во овој пример (поради неговата природа) кон еден јазел покажува само една врска, односно постои некаква хиерархија. Речиси секоја општествена организација тргнувајќи од мало претпријатие па до државата подлежи на хиерархиска организација. Структурите во компјутерската техника се многу често хиерархиски: базите на податоци, синтаксичките структури во програмските јазици, формулите итн.

Постојат и уште покомплексни структури кои дозволуваат повеќе врски да покажуваат (односно да делат) еден јазел. Примери за такви структури се дадени на слика 5.2. Овие структури би можеле да се искористат за моделирање на електронски книги или веб страници кои дозволуваат користење на така наречени хиперврски (hyperlinks).



(а) сложена листа каде подлистите се заеднички



(б) листа со јамка

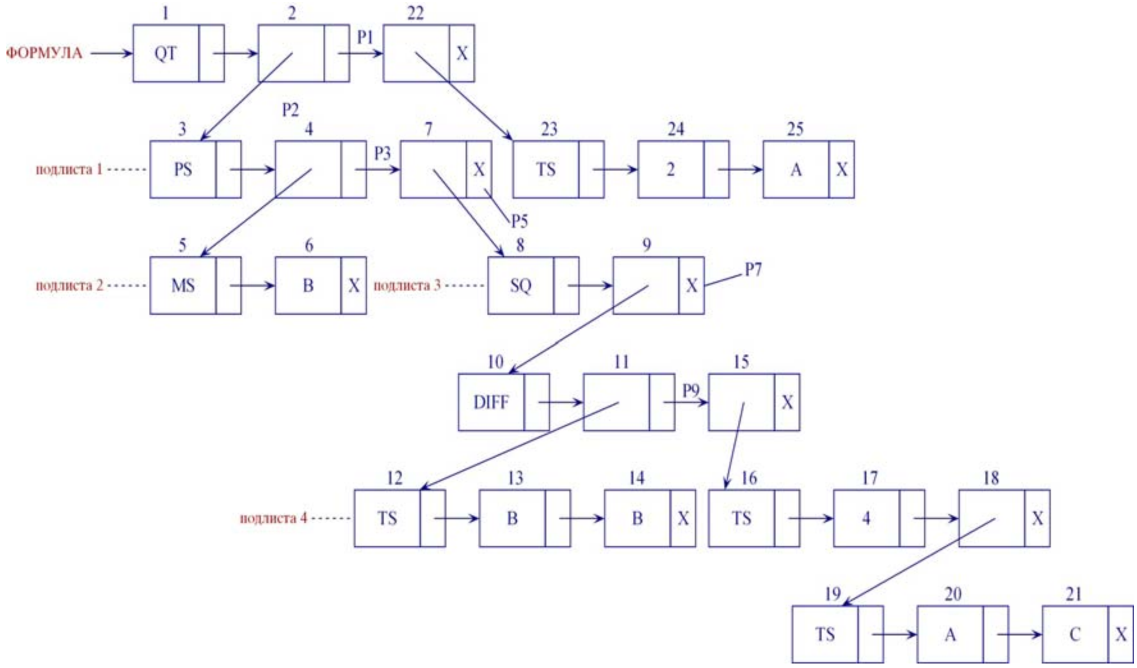
слика 5.2 Нехиерархиски комплексни листи

Најважни операции кои се извршуваат со листите се: вметнување, бришење и изминување на јазлите од листите. Операциите вметнување и бришење обично ја содржат и операцијата изминување за да овозможат одговор на прашањето: каде да се вметне елементот односно кој елемент да се избрише? (За потсетување: При едноставните листи, специјалните случаи на бришење односно вметнување елементи на почетокот или крајот на листата дефинираат специјални податочни типови)

Во ова поглавје од интерес ќе бидат хиерархиските сложени листи. Нехиерархиските сложени листи ќе бидат обработени во темата графови.

Изминувањето на хиерархиските сложени листи може да се опише на следниот начин:

1. Пристапи до првиот јазел (доколку постои)
2. Процесирај го јазелот до кој си пристапил
3. Доколку јазелот е комплексен, измини ја листата (односно листите) кон која тој покажува.
4. Пристапи кон следниот јазел (доколку постои)



слика 5.3 Илустрација на операцијата изминување на комплексна листа

Овој алгоритам е рекурзивен, но неговата имплементација може, но не мора да биде рекурзивна. За илустрација, изминувањето на една комплексна листа е дадено на слика 5.3. При тоа, јазлите од комплексната листа се означени со броеви според редоследот на нивното изминување. Може да се забележи дека изминувањето на јазлите е во основа слично со реализирањето на повиците на функциите при извршувањето на една програма (види го поглавјето кое се однесува на динамички структури - листи). Од тука следи дека нерекурзивното изминување на ваквата комплексна листа ќе инволвира користење на магацин во кој би се чувале покажувачите кон следниот елемент од листата секогаш кога се преминува на изминување на подлистата на некој тековен јазел. По изминувањето на подлистата, се продолжува со изминувањето на првичната листа од местото кое е одредено со покажувачот кој се вади од магацинот. Целото изминување завршува кога магацинот ќе остане празен.

5.2 Податочни структури: Стебла

Стеблата во најопшта форма претставуваат хиерархиска колекција на елементи. Со нив се претставуваат хиерархиски структури и како такви се една од најважните нелинеарни структури што се сретнуваат во компјутерските алгоритми. Не секоја хиерархиска структура е стебло, но секоја хиерархиска структура може да биде претставена во облик на стебло.

Интуитивно, под нелинеарна структура “стебло” се подразбира групирање на објекти преку гранење слично на стеблата во природата. Стебло е колекција од елементи наречени јазли, од кои еден јазел се нарекува корен. За оваа колекција е дефинирана релацијата „е родител”. Во секој од јазлите на стеблото може да се сместат податоци од кој било податочен тип.

Поформално, едно стебло може да се опише со следната дефиниција:

1. Јазел сам за себе претставува стебло. Тогаш, јазелот е и корен на стеблото.
2. Нека n е јазел и T1, T2, ..., Tk се стебла со корени n1, n2, …, nk соодветно. Тогаш стебло може да се конструира ако јазелот n го направиме корен на стеблото што ги содржи подстеблата T1, T2, ..., Tk. Јазлите n1, n2, …, nk ги нарекуваме деца на јазелот n.

Коренот на секое стебло е единствен јазел во стеблото што нема родител. Јазел кој има барем еден наследник се нарекува нетерминален (внатрешен) јазел. Ако еден јазел нема наследници, тогаш истиот се нарекува терминален јазел или лист.

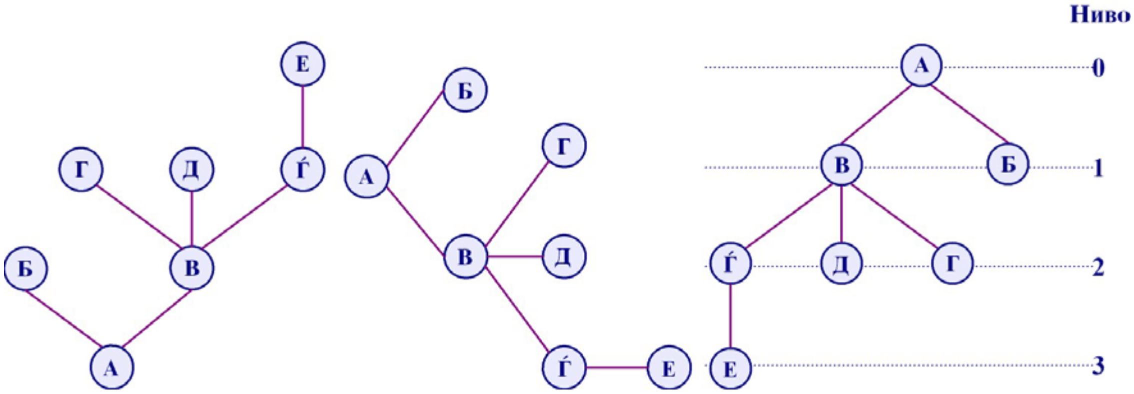
Стеблата можат да се дефинираат рекурзивно на следниот начин:

Стебло е конечно множество T со еден или повеќе елементи наречени јазли што ги задоволува следниве правила:

1. Постои еден јазел наречен корен на стеблото;
2. Останатите јазли (без коренот) се групирани во $k \geq 0$ дисјунктни множества T1, T2, ..., Tk, од кои секое е стебло. Овие стебла се нарекуваат подстебла на стеблото T.

Постојат и други начини на дефинирање на стебло кои обично се базираат на теоријата на вгнездени множества или графови.

На слика 5.4 се прикажани три начини на претставување на едно стебло.

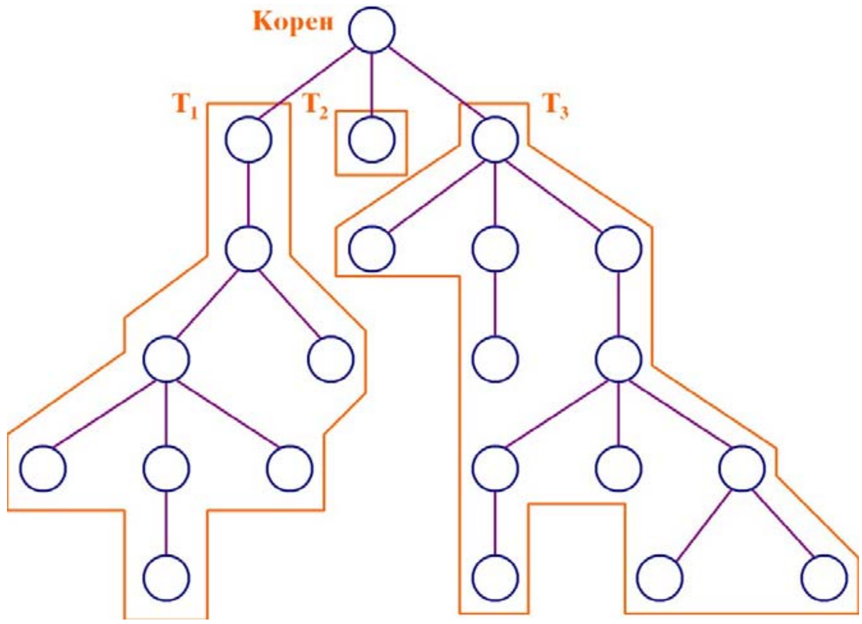


слика 5.4. Три претстави на едно стебло

Јазелот А е корен (root) на стеблото. Јазелот В е корен на подстеблото {Г,Д,Ѓ,Е}. Трите скици претставуваат исто стебло и се добиени со едноставни ротации. Во пракса најчесто се користи третиот дијаграм, каде што стеблото се претставува од горе (коренот е горе) кон долу.

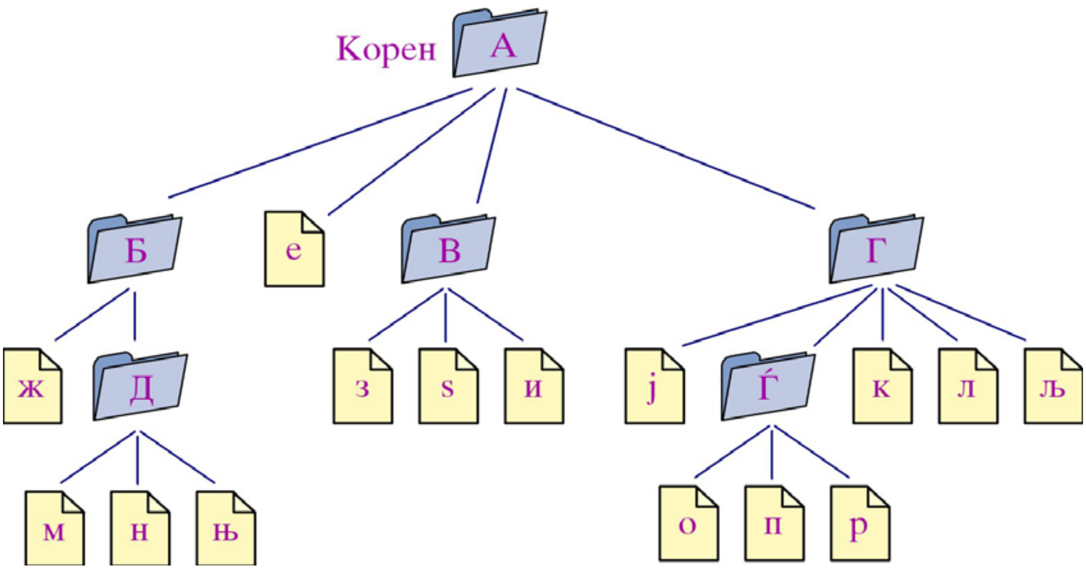
Од дефиницијата за стебло следува дека секој внатрешен јазел во стеблото е корен на некое подстебло (види слика 5.5). Бројот на подстебла на еден јазел се нарекува степен на јазелот. Кога овој број е 0, јазелот се нарекува краен (терминален) јазел или лист. Сите јазли (освен коренот) имаат свој родител.

Јазел што има најмалку едно дете се нарекува нетерминален или внатрешен јазел. Нивото на секој јазел е бројот на подстебла преку кои се доаѓа до него, односно бројот на јазли низ кои треба да се помине за од коренот да се стигне до јазелот, имајќи предвид дека нивото на коренот е 0. Стандардната терминологија за јазлите во стеблата е земена од фамилијарните стебла. Така, секој корен е татко (или родител). Јазлите непосредно под таткото се негови деца (понекогаш се користат и термините синови или ќерки), а меѓу себе се браќа (сестри). Некои автори ги користат термините наследници и следбеници за да го претставаат односот на родителство.



слика 5.5. Корен на стебло со подстебла

На слика 5.6, е прикажано стебло, во кое за поедноставно објаснување јазлите се означени со букви од кириличната азбука. Според сликата јазелот означен со Г е родител на јазлите Ѓ, ј, к, л, љ. Јазелот означен со Ѓ е родител на јазлите о, п, р. Јазлите Д и ж се деца на јазелот Б. Сите јазли означени со буквите меѓу е и р се листови на стеблото.



слика 5.6. Стебло

Множество (обично подредено) на различни (дисјунктни) стебла се нарекува шума. Ако од едно стебло го избришеме коренот, се добива шума. Ако пак во една шума додадеме само еден јазел и го поврземе со корените на стеблата, од шумата добиваме едно стебло.

Ако n1, n2, …, nk е низа на јазли во стебло така да ni е родител на ni+1, 1 ≤i < n, тогаш низата се нарекува патека од јазелот n1 до nk. За слика 5.6, патеката од јазелот А до м гласи: А, Б, Д, м.

Должина на патека претставува број на врски меѓу два јазла, односно е за еден помала од бројот на јазли во патеката. Должина на патека од еден јазел до самиот себе изнесува 0.

Ако постои патека од јазелот означен со А во некое стебло до јазелот означен со В, тогаш велиме дека А е предок на В, а В е наследник на А. За слика 5.6 важи, наследници на јазелот Б се јазлите Д, ж, м, н, њ. Секој јазел е предок и наследник на самиот себе. Предци на еден јазел се сите јазли долж патеката од коренот до самиот јазел. За стеблото на слика 5.6, претци на јазелот м се јазлите А, Б, Д.

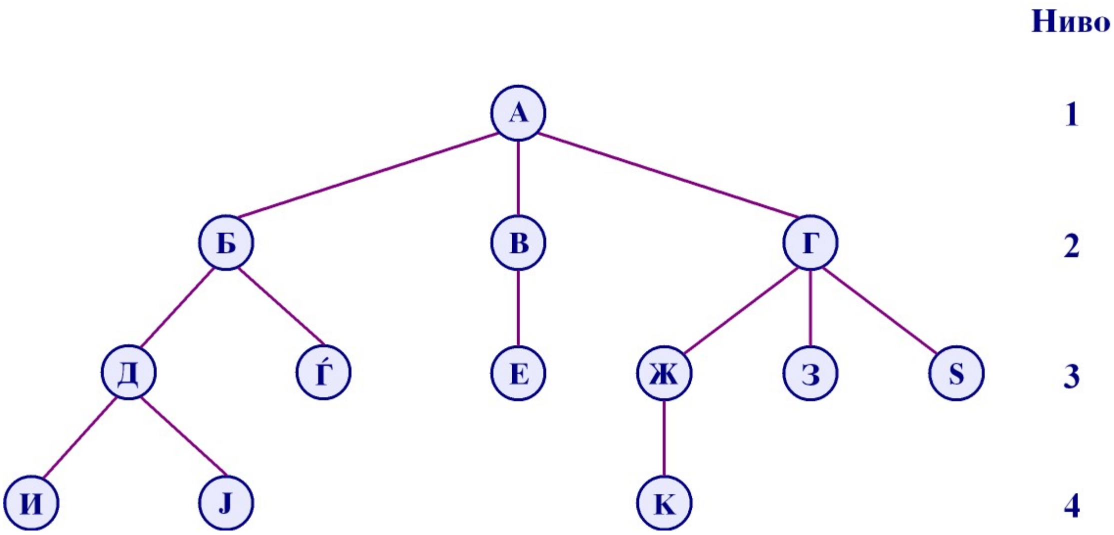
Вистински наследник (предок) на даден јазел А е јазел кој е различен од јазелот А. Јазел без вистински наследник се нарекува лист. За стеблото на слика 5.6 терминалните јазли се означени со малите букви од кириличната азбука.

Подстебло на даден јазел во стебло е јазелот дете со сите свои наследници. Бројот на подстебла на еден јазел се нарекува степен на стеблото. За стеблото на слика 5.6, јазелот А има степен 4, јазелот Г 5 итн. Степен на стебло е максималниот степен на јазлите во стеблото. За стеблото на слика 5.6, степенот на стеблото изнесува 5.

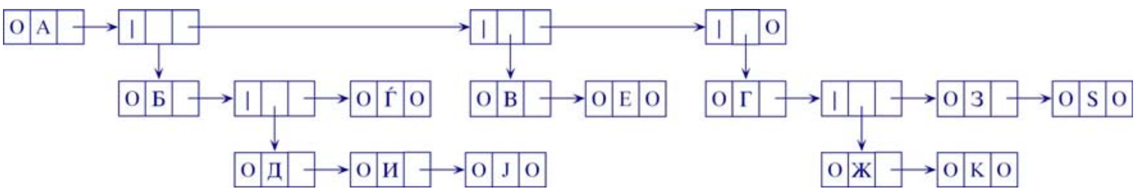
Висина на јазел во стебло е должината на најдолгата патека од јазелот до листовите. Висина на стебло е висина на коренот на стеблото. Длабочина на јазел е должината на единствената патека од коренот до јазелот. Длабочина на стебло е дефинирана со максималната длабочина на било кој јазел од стеблото.

5.3 Репрезентација на стеблата

Најприродно претставување на стебло е едноставно родителот да ги содржи покажувачите кон своите деца. Ваквото "идеално" претставување содржи проблеми бидејќи еден јазел може да има произволен број деца. Така бројот на покажувачи во јазлите би требало да биде променлив или пак би требало да се предвиди некој максимален број на деца и секој јазел да има простор за таков број на покажувачи. Поради оваа неудобност, стеблата обично се репрезентираат со структура каде што бројот на покажувачи е секогаш два. Едно вакво претставување на бинарно стебло е прикажано на слика 5.7.



слика 5.7а. Пример Стебло



слика 5.7б. Репрезентација на пример стеблото со структура со два покажувачи

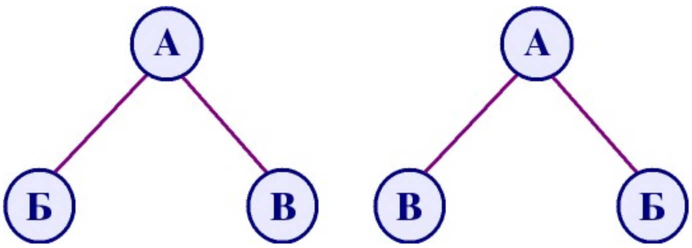
Стеблото прикажано на слика 5.7б има интересни својства. Јазелот на едно такво стебло може да биде даден со следниот код:

```
typedef int info_t;
typedef struct jazel
{
      info_t info;
      struct jazel *left;
      struct jazel *right;
} node;
```

При што info_t може да биде било каков запис.

Ова претставување на стеблата се нарекува претставување со помош на бинарни стебла. Бинарното стебло е погодно за најголемиот дел од алгоритмите кои работат со стебла (односно со хиерархии на податоци). Единствен недостаток е значителниот број на празни покажувачи (означени со вредност нула). Понатаму ќе видиме како овие слободни места за покажувачи можат корисно да се употребат.

Кога редоследот на подстеблата во едно стебло е важен, стеблото се нарекува подредено стебло. На примерот од слика 5.8 , следниве две подредени стебла се различни:



слика 5.8. Подредено стебло

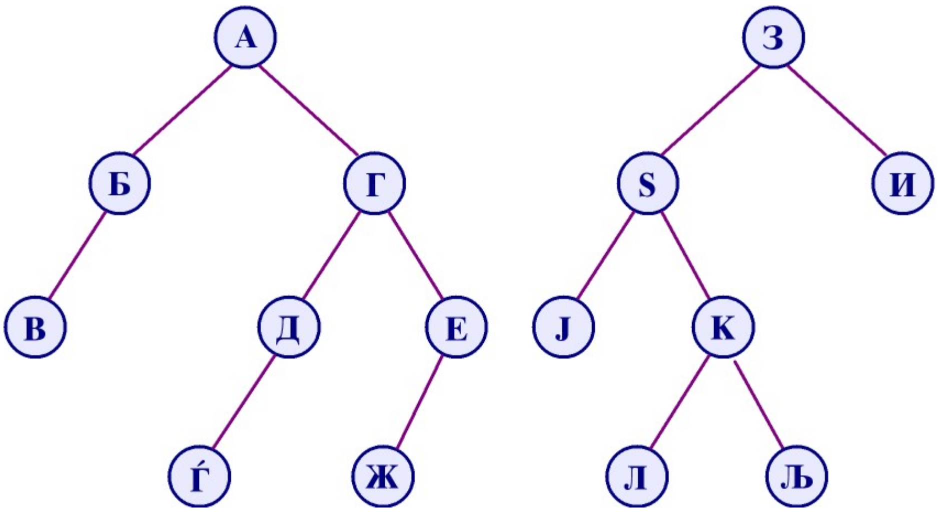
За подредените стебла важи: ако a и b се деца на јазелот T и ако a е лево, а b десно, тогаш и сите потомци на a се лево од b.

5.4 Бинарни стебла

Бинарните стебла се стебла кај кои секој јазел може да има најмногу две подстебла, при што ако постои само едно од нив ние разликуваме дали е левото или десното. Бинарно стебло е стебло чиј степен изнесува два, односно сите негови јазли може да немаат наследници, да имаат еден или најмногу два наследници. Кај бинарните стебла разликуваме лево и десно подстебло.

Поформална дефиниција на бинарното стебло е дадена во продолжение. Бинарно стебло е конечно множество на јазли кое е или празно или се состои од корен и две дисјунктни подстебла наречени лево и десно подстебло.

На слика 5.9 се прикажани две бинарни стебла. Стеблото на десната страна репрезентира комплетно бинарно стебло. Комплетно бинарно стебло е стебло чии нетерминални јазли секогаш имаат два потомка.

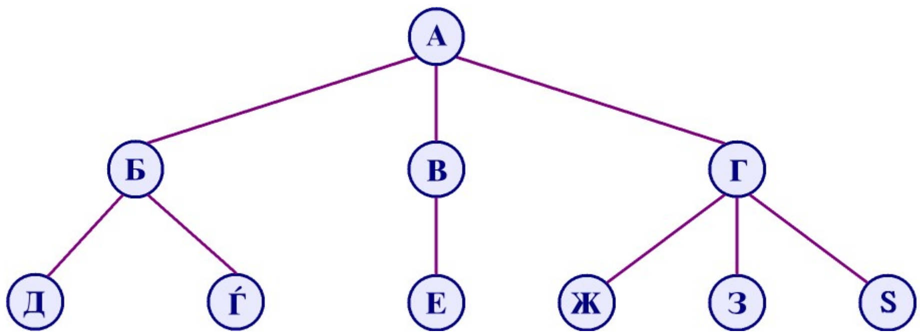


слика 5.9. Некомплетно и комплетно бинарно стебло

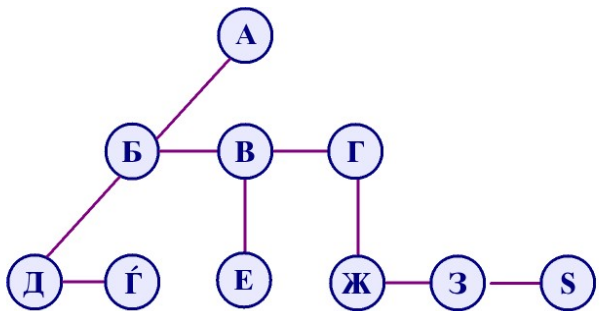
Класата на бинарни стебла не е подкласа на класата стебла туку нов поим иако двете класи се со многу сличности. Во суштина двете клучни работи што ги прават бинарните стебла посебна класа се, правењето разлика помеѓу лево и десно подстебло и можноста бинарното стебло да биде празно. Важноста на бинарните стебла не потекнува само од нивната практична применливост туку и од фактот што секое стебло на едноставен начин може да се претстави како бинарно стебло.

5.5 Трансформација на шума од произволни стебла во бинарно стебло

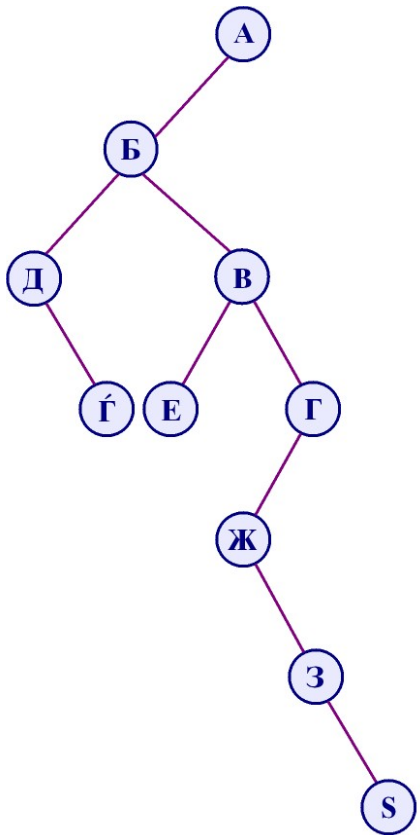
Откако поимите стебло и бинарно стебло се дефинирани, може да се опише и алгоритмот за претставување на произволно стебло (односно шума од стебла) со помош на бинарно стебло. Стеблото дадено на слика 5.10а може да се измине така што се изминува коренот, па сите негови деца, при што таа постапка рекурзивно се повторува. Таквото изминување е прикажано на сликата 5.10б, која доколку се заврти за 45 степени ја дава сликата 5.10в со која е претставено бинарното стебло кое соодветствува на стеблото од слика 5.10а.



слика 5.10а. Произволно стебло

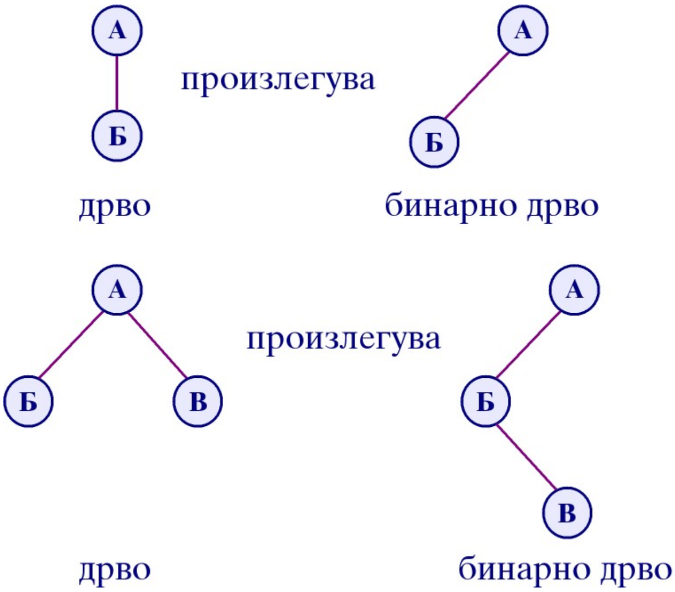


слика 5.10б. Изминување на произволното стебло според хиерархијата е дете



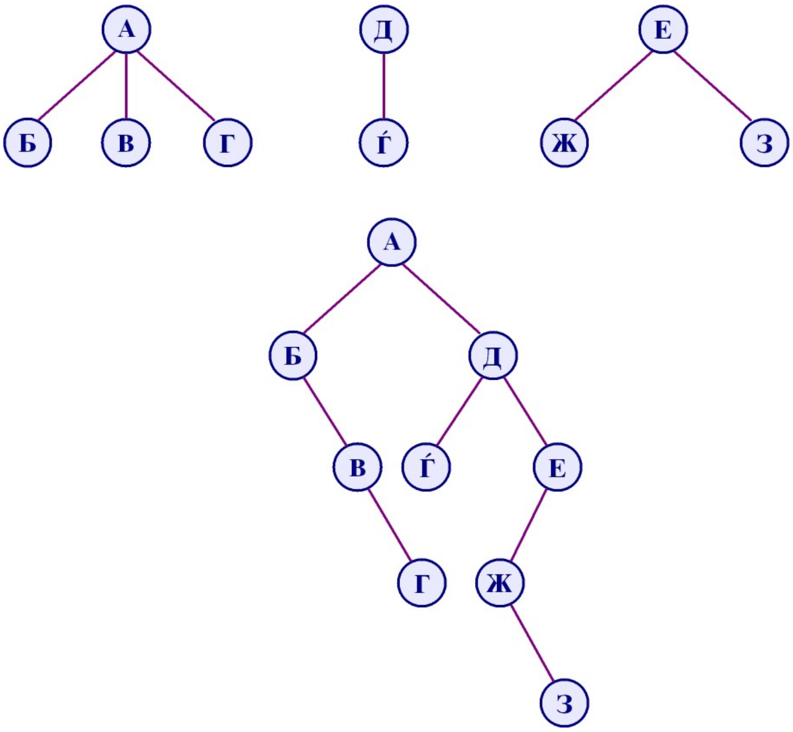
слика 5.10в. Бинарно стебло репрезент на стеблото од слика 5.10а

Оваа трансформација важи за произволни стебла. На слика 5.11 е претставена примената на таа трансформација за две едноставни стебла.



слика 5.11. Репрезентација на едноставни стебла со бинарно стебло

Репрезентацијата на шума од стебла со помош на бинарно стебло е тривијална ако се претпостави дека корените на сите стебла од шумата се браќа. Тогаш согласно предходно кажаното за даден пример се добива резултатот од слика 5.12.



слика 5.12 Репрезентација на шума од стебла со бинарно стебло

Предходно илустрираната трансформација на произволно стебло во бинарно стебло може да се опише поформално на следниот начин:

Нека T1, ...,Tn е шума на стебла, тогаш бинарното стебло кое се добива со трансформацијата на оваа шума може да се означи со B(T1, ..., Tn) и за неа важи:

B(T1, ..., Tn)

(i) е празно ако n = 0;

(ii) има корен еднаков со коренот (T1); има лево подстебло B(T11,T12, ...,T1m) каде T11, ..., T1m се подстебла на коренот (T1); има десно подстебло B(T2, ..., Tn).

5.6 Изминување на бинарните стебла

Постојат три основни начини на кои може да се изминат сите елементи на едно бинарно стебло: преордер, инордер и постордер изминување.

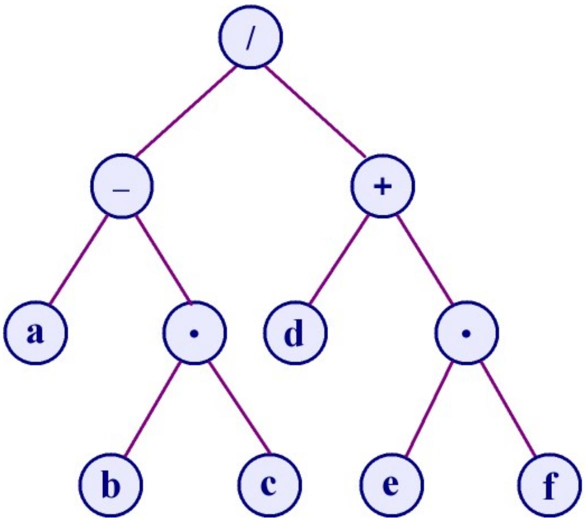
Ако T е празно стебло, тогаш празна листа се добива при преордер, инордер и постордер изминување на стеблото. Ако T содржи еден јазел, тогаш тој јазел претставува инордер, постордер и преордер изминување на стеблото T. Нека ја имаме структурата на стеблото чиј корен е јазелот K и ги содржи подстеблата T1 и T2. Тогаш:

Преордер изминување на јазлите на T е низа која ја сочинува информацијата од коренот K на кој му следи низата од информациите од јазлите на подстеблото T1 изминато во преордер, по кое следи низата од информациите од јазлите од подстеблотоT2 во преордер.

Инордер изминување на јазлите на T е низата од информациите од јазлите на подстеблотоT1 изминато во инордер, потоа информацијата од јазелот K, на кои им следи низата од информациите од јазлите од T2 изминато во инордер.

Постордер изминување на јазлите на T е низа која ја сочинува низата од информациите од јазлите на подстеблотоT1 изминато во постордер, потоа низата од информациите од јазлите од подстеблотоT2 изминато во постордер и на крајот информацијата од коренот.

Една од најчестите примени на овие изминувања е во пресметувањето на аритметичките изрази со помош на компјутерите (или дигитроните). Аритметичките изрази обично имаат еден или два операнда, со што претставуваат добар пример за можност на користење на бинарно стебло за нивна репрезентација. Во компјутерската анализа на аритметичките (и логичките) изрази важна улога играат начините на поминувањето на стеблото со кои се претставени тие изрази. На слика 5.13 е дадено стебло за еден аритметички израз и соодветните поминувања на стеблото. Стеблото го претставува изразот: (a - b · c)/(d + e · f)



слика 5.13. Стебло на аритметички израз

Изминувањето на стеблото ги дава следните низи:

преордер: · - a · bc + d · e f

инордер: a - b · c / d + e · f

постордер: a b c · - d e f · + /

Во литературата овие поминувања на стеблото претставени во соодветните низи на информации од јазлите во англиската терминологија се нарекуваат: prefix, infix и postfix нотација и како такви се користат во реализацијата на преведувачите на програмските јазици.

Алгоритмите за основните начини на поминување на стебло во својата рекурзивна верзија се речиси тривијални:

```
void inorder(nodep p)
{
 if(p!=NULL)
 {
  inorder(p->left);
  printf("%d\n",p->info);
  inorder(p->right);
 }
}
```

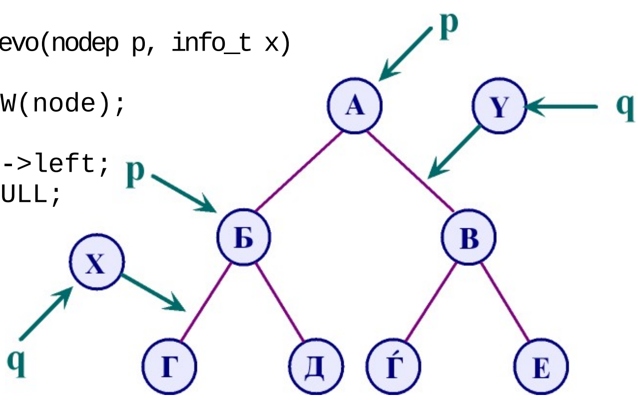
```
void preorder(nodep p)
{
 if(p!=NULL)
 {
  printf("%d\n",p->info);
  preorder(p->left);
  preorder(p->right);
 }
}
```

```
void postorder(nodep p)
{
 if(p!=NULL)
 {
  postorder(p->left);
  postorder(p->right);
  printf("%d\n",p->info);
 }
}
```

5.7 Основни манипулации со јазлите во бинарните стебла

Вистинската вредност на динамичките структури е во можноста за едноставно менување на структурата во смисла на вметнување нови и/или бришење на елементи. За бинарните стебла динамичноста е особено важна бидејќи нивното користење во практичните алгоритми најчесто е базирано на нивно постојано менување. Понатаму ќе ги разгледаме двете основни операции на менување на стеблото, вметнување и бришење јазел. На слика 5.14 е дадена илустрација на вметнување јазел во бинарно стебло заедно со соодветните алгоритми.

```
void vmetni_levo(nodep p, info_t x)
{
 nodep q=NEW(node);
 q->info=x;
 q->left= p->left;
 q->right=NULL;
 p->left=q;
}
```




```
void vmetni_desno(nodep p, info_t x)
{
 nodep q=NEW(node);
 q->info=x;
 q->right = p->right;
 q->left=NULL;
 p->right=q;
}
```

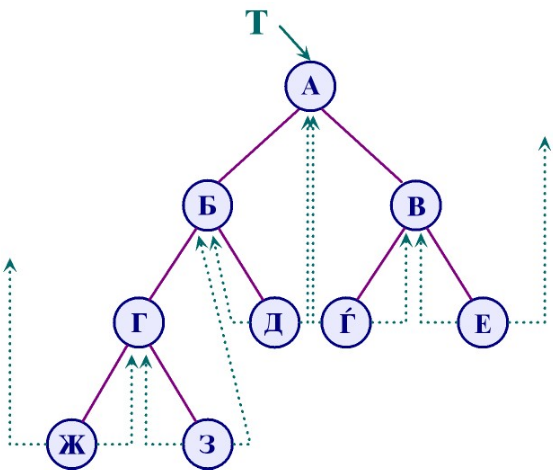
слика 5.14. Вметнување јазел со покажувач q под јазел со покажувач p во лево и десно подстебло од едно бинарно стебло

Бришење јазел во стебло е посложена операција, во случаите кога тој има две деца. Проблемот е во тоа што со еден влезен покажувач во јазелот не можеме да ги покриеме двата излезни покажувачи кон неговите деца. Во таквите случаи, избришаниот јазел треба да биде заменет со друг јазел (најчесто некој од неговото подстебло). На пример, кај бинарните стебла на пребарување, при бришењата ваквите јазли се заменуваат со најдесниот јазел од левото подстебло. Кога јазелот што се брише нема деца или пак има само едно дете, алгоритамот е тривијален.

5.8 Поврзани бинарни стебла

Бинарната репрезентација на стеблата резултира во голем број на неискористени покажувачи. Овие покажувачи можат да се искористат така што тие ќе претставуваат специјална врста на покажувачи односно таканаречени нишки (treads). Нишките се покажувачи кој покажуваат на предходникот односно следбеникот на даден јазел во некое изминување. При тоа се зема дека нишката покажува на предходникот доколку е тоа неискористена лева врска на јазелот. Соодветно доколку нишката е неискористена десна врска на јазлеот, таа покажува во наследникот на јазелот во избраната нотација.

На тој начин бинарните стебла можат да бидат нанижани во преордер, постордер и инордер нотацијата. Најчесто се сретнуваат инордер нанижани бинарни стебла. Пример за такво инордер нанижување на бинарно стебло е даден на слика 5.15. За да се разликуваат нишките од правите врски (линкови) тие се означени со испрекинати линии.



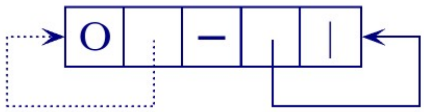
слика 5.15 Инордер нанижано бинарно стебло

Од сликата може да се забележи дека првиот и последниот јазел во избраното изминување не покажуваат кон јазел од стеблото, додека сите други врски во стебото се искористени. На овој начин се поедноставуваат процедурите за изминување на стеблото. (Алгоритмите за нерекурзивно изминување на ненанижано и нанижано стебло ќе бидат дадени на аудиториските вежби).

Од дефиницијата на бинарното стебло се дозволува тоа да биде празно. Поради поедноставна програмска имплементација на изминувањето на стеблото (како да се знае кога да се застане) и воедно искористување на двете неискористени врски може да се воведе јазел водач (кој покажува кон коренот на стеблото), а кон кој ќе покажуваат двете неискористени врски. Втор проблем кој треба да се реши со воведувањето на нишките е начинот на кој програмот кој работи со стеблото ќе прави разлика од придвижувањето по врска од стеблото или по нишка од стеблото (и двете се реализирани со ист покажувач). Наједноставно е да се воведе дополнително поле за секој од двата покажувача со кое ќе се опише дали покажувачот опишува врска или нишка. Во тој случај освен инфо (INFO) полето, и два покажувача (LLINK, RLINK) јазелот содржи и две полиња за ознака (најчесто означени со LTAG и RTAG односно LBIT и RBIT). При тоа вообичаено се зема дека доколку RBIT е ’1’ станува збор за десна врска, додека ’0’ означува дека станува збор за десна нишка од даден јазел.

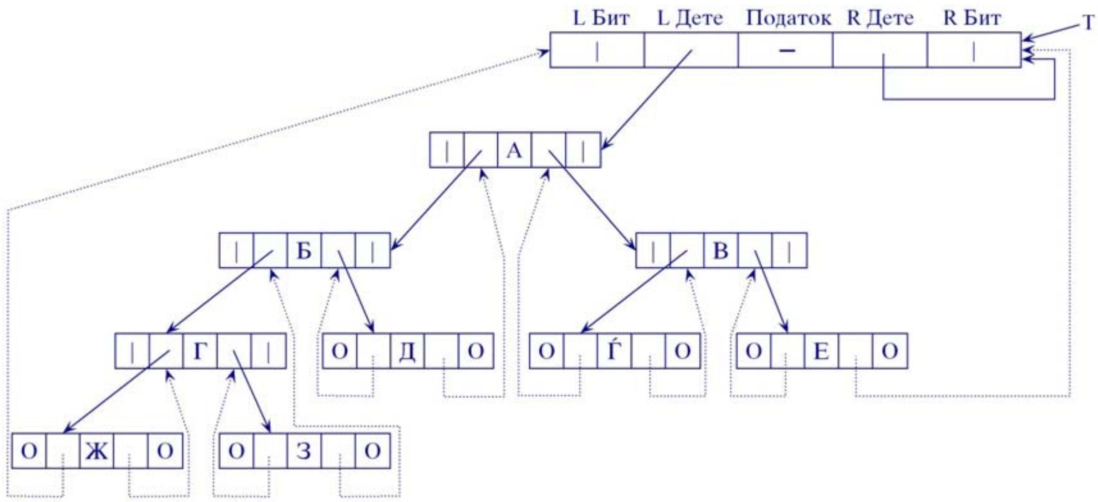
```
typedef int info_t;
typedef struct jazel
{
       info_t info;
       struct jazel *left;
       int lbit;
       struct jazel *right;
       int rbit;
} node;
```

Празното бинарно стебло може да се прикаже како на слика 5.16. Треба да се забележи дека едната од врските на бинарното стебло треба да постои.



слика 5.16 Репрезентација на празно нанижано бинарно стебло

Комплетна репрезентација на нанижано бинарно стебло е дадена на слика 5.17. Важно е да се напомене дека иако операциите на изминување на овој начин стануваат полесни за имплементација, операциите на внесување, односно бришење на јазел стануваат покомплицирани поради тоа што не треба да се наруши правилната нанижаност. Повеќе детали за овие операции ќе бидат дадени на аудиториските вежби.



слика 5.17 Репрезентација на нанижано бинарно стебло